AD-A106 740   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH   F/G 20/11
A FINITE ELEMENT MODEL OF A WHITE-METZNER VISCOELASTIC POLYMER --ETC(U)
FEB 81   B R COLLINS
UNCLASSIFIED   AFIT-CI-81-31T   NL

1 of 2
AD
A106740

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 81-31T (6)
2. GOVT ACCESSION NO.: AD-A106740
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): A Finite Element Model of a White-Metzner Viscoelastic Polymer Extrudate (1)
5. TYPE OF REPORT & PERIOD COVERED: THESIS/~~DISSERTATION~~/
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): (10) Brent R./Collins (14) AFIT-CI-81-31T
8. CONTRACT OR GRANT NUMBER(s): (9) Master's thesis
9. PERFORMING ORGANIZATION NAME AND ADDRESS: AFIT STUDENT AT: Massachusetts Institute of Technology
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: (11) (12) 135
11. CONTROLLING OFFICE NAME AND ADDRESS: AFIT/NR WPAFB OH 45433
12. REPORT DATE: Feb 1981
13. NUMBER OF PAGES: 82
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): LEVEL II
15. SECURITY CLASS. (of this report): UNCLASS
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report): APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DTIC SELECTE NOV 6 1981 D S H

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

20 OCT 1981 Fredric C. Lynch

FREDRIC C. LYNCH, Major, USAF
Director of Public Affairs
Air Force Institute of Technology (ATC)
Wright-Patterson AFB, OH 45433

18. SUPPLEMENTARY NOTES: APPROVED FOR PUBLIC RELEASE: IAW AFR 190-17
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
20. ABSTRACT (Continue on reverse side if necessary and identify by block number): ATTACHED

012200

JOB

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A FINITE ELEMENT MODEL

OF A WHITE-METZNER VISCOELASTIC

POLYMER EXTRUDATE

by

BRENT R. COLLINS

Submitted to the Department of Aeronautics and Astronautics on January 16, 1981 in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

A finite element model of a viscoelastic polymer melt characterized by a White-Metzner rheological equation of state was developed. For creeping flow wherein the inertia terms are negligible non-linear finite element equations were solved by a method of direct substitution termed Picard iteration.

Four flow geometries were examined: cross channel, plane couette, entry, and step flow. A comparison of two bi-quadradic isoparametric element types (8 node "serendipity" and 9 node "Lagrange") showed general superior behavior of the Lagrange elements. The "penalty" method of incompressible flow was used with the Galerkin method to formulate the finite element equation, yielding satisfactory behavior for creeping inelastic and viscoelastic flow.

The non-linear equations yielded numerical convergence up to Weissenberg numbers of 0.01. Techniques of expanding this radius of convergence were examined and proposed for future effort.

Thesis Supervisor: Dr. David K. Roylance

Title: Associate Professor of Materials Engineering

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to ascertain the value and/or contribution of research accomplished by students or faculty of the Air Force Institute of Technology (ATC). It would be greatly appreciated if you would complete the following questionnaire and return it to:

AFIT/NR
Wright-Patterson AFB OH 45433

RESEARCH TITLE: A Finite Element Model of a White-Metzner Viscoelastic Polymer Extrudate

AUTHOR: Brent R. Collins

RESEARCH ASSESSMENT QUESTIONS:

1. Did this research contribute to a current Air Force project?

( ) a. YES ( ) b. NO

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not?

( ) a. YES ( ) b. NO

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency achieved/received by virtue of AFIT performing the research. Can you estimate what this research would have cost if it had been accomplished under contract or if it had been done in-house in terms of manpower and/or dollars?

( ) a. MAN-YEARS __________ ( ) b. $__________

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3. above), what is your estimate of its significance?

( ) a. HIGHLY SIGNIFICANT ( ) b. SIGNIFICANT ( ) c. SLIGHTLY SIGNIFICANT ( ) d. OF NO SIGNIFICANCE

5. AFIT welcomes any further comments you may have on the above questions, or any additional details concerning the current application, future potential, or other value of this research. Please use the bottom part of this questionnaire for your statement(s).

NAME GRADE POSITION

ORGANIZATION LOCATION

STATEMENT(s):

81 10 26 182

FOLD DOWN ON OUTSIDE - SEAL WITH TAPE

AFIT/NR
WRIGHT-PATTERSON AFB OH 45433

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE. $300

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 73236 WASHINGTON D.C.

POSTAGE WILL BE PAID BY ADDRESSEE

AFIT/ DAA
Wright-Patterson AFB OH 45433

FOLD IN

8 1-31T

A FINITE ELEMENT MODEL

OF A WHITE-METZNER VISCOELASTIC

POLYMER EXTRUDATE

by

BRENT R. COLLINS

SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE

DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1981

Signature of Author Brent R. Collins
Brent R. Collins, Department of Aeronautics and Astronautics

Approved by Roy A. Schluntz
Roy A. Schluntz, Technical Supervisor, CSDL

Certified by David K. Roylance
David K. Roylance, Thesis Supervisor

Accepted by Harold Y. Wachman
Harold Y. Wachman, Chairman, Departmental Graduate Committee

A FINITE ELEMENT MODEL

OF A WHITE-METZNER VISCOELASTIC

POLYMER EXTRUDATE

by

BRENT R. COLLINS

Submitted to the Department of Aeronautics and Astronautics on January 16, 1981 in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

A finite element model of a viscoelastic polymer melt characterized by a White-Metzner rheological equation of state was developed. For creeping flow wherein the inertia terms are negligible non-linear finite element equations were solved by a method of direct substitution termed Picard iteration.

Four flow geometries were examined: cross channel, plane couette, entry, and step flow. A comparison of two bi-quadradic isoparametric element types (8 node "serendipity" and 9 node "Lagrange") showed general superior behavior of the Lagrange elements. The "penalty" method of incompressible flow was used with the Galerkin method to formulate the finite element equation, yielding satisfactory behavior for creeping inelastic and viscoelastic flow.

The non-linear equations yielded numerical convergence up to Weissenberg numbers of 0.01. Techniques of expanding this radius of convergence were examined and proposed for future effort.

Thesis Supervisor: Dr. David K. Roylance

Title: Associate Professor of Materials Engineering

TABLE OF CONTENTS

LIST OF FIGURES

TABLE

ACKNOWLEDGEMENTS

I am most grateful to Professor David K. Roylance for his timely guidance of my work.

I would also like to acknowledge the financial support of the Charles Stark Draper Laboratory which gave me the opportunity to pursue this project and provided funding for the computer time under IR & D Contract No. 18339. I also wish to acknowledge the sponsorship of the United States Air Force under the direction of the Air Fcrce Institute of Technology.

I am pleased to acknowledge "Kid" Benny Fudge, Billy White, and Mitch Hansberry who gave me no help at all.

I. INTRODUCTION

The finite element method has become a popular numerical tool in the analysis of fluid flow problems (1, 2, 3, 4, 5). Particularly in the regime of incompressible flow this method has become very competitive with the more established finite difference methods due to its simplicity of implementation and generality of handling mixed boundary conditions for complex geometries which favor nonuniform meshes at points of singularities. Accordingly, the finite element discretization process is used herein to characterize the flow of a polymeric melt under various geometric conditions. The particular approach is the Galerkin weighted residual formation of the non-symmetric integral equations (6,2), with the penalty method used for the pressure term via an approximation of the incompressible continuity equation (1,3,7). Steady, two dimensional flow is treated and viscoelastic fluid effects are modeled by employing an Oldroyd codeformational stress derivative in a modified Maxwell constitutive equation (8).

The motivation for this analysis stems from the development of low cost, medium performance, plastic gyroscopic instruments at the Charles Stark Draper Laboratory. With the exception of the momentum wheel and electromagnetic parts, a complete single degree of freedom integrating gyroscope has been designed using glass filled polyphenylene sulfide parts. Performance goals are in the range of 1 - 10 degrees/hour drift rate. Cost advantages derive from elimination of precision secondary machining

of metallic components as well as basic material costs. However, the need for uniform physical and mechanical properties (e.g., dimensional stability, thermal conductivity, mechanical compliance) of the parts to provide the required performance after possible long-term storage in unair-conditioned warehouses necessitates the correlation of the final material state to processing parameters. Such knowledge will permit the rationale selection of extrusion parameters, post fabrication treatments, and subsequent analysis of storage and service environmental effects on instrument performance. Figure 1 shows a picture of the typical plastic gyroscope under consideration. Figure 2 depicts a typical injection molding process for these gyroscopic parts.

Roylance (9) has pointed out that the information the engineer is seeking in a flow analysis is the location of regions of elevated shear deformation, which can lead to mechanical degradation and higher residual stresses, regions of stagnation and recirculation, at which overlong material residence and thermal degradation might occur, and power requirements for the fabrication process itself. Also of interest for the gyroscope application are the effects of the flow field on the distribution of the filler fibers which are carried along by the drag of the fluid. It is possible that the zones of filler depletion or enhancement which are observed in molded parts, can be predicted and controlled by evaluation of the calculated velocity field.

In the above regard the numerical analysis of polymer melts can be broken down into two general categories. First is the evaluation of the accuracy of the solution themselves. Calculations are made and compared to known exact or approximate analytic solutions. Typical of these are the pipe/channel flow, drag flow, and die entry flow. By far most of the numerical studies have been in this category. In the second category is the application of numerical solutions to real problems. Only three studies are known to this author which have aimed at applying numerical results to actual polymer processing. The first is the work of Bigg (10) who used the Marker and Cell Finite Difference Scheme to specify preferred operations for the mixing of polymer solutions in a single screw extruder. The second is the National Science Foundation/Industry supported work at Cornell University (10), also using finite difference methods to evaluate mold filling and control the location and orientation of weld lines. Thirdly is the work of Caswell and Tanner (12) who effectively used the finite element method to redesign wire coating dies to eliminate recirculation.

The current work falls into the first category described above, but the intention of applying the numerical model, once assessed for accuracy and utility, is kept firmly in mind and discussed throughout this report. To conclude the introduction, it is also necessary to describe how the current analysis fits into the completely general solution. In the injection molding process, the flow is non-steady and non-isothermal (but approximately adiabatic within the fluid boundaries), with advancing free surfaces until the mold is completely filled. Upstream of

the flow front the fluid is completely surrounded by either rigid boundaries or adjacent fluid. For an incompressible fluid, a complete numerical model must therefore account for unsteady, non-isothermal, free surface effects. In addition, the observance of a finite recoverable shear in the rheological data of polymer melts indicates the need to include viscoelastic effects in the model. For unsteady effects, since the Reynolds number (Re) of flow is always much less than unity, a good approximation is achieved by ignoring inertia and employing the linear "creeping" flow solution. The model that we are eventually striving for then is an adiabatic, viscoelastic solution with changing surface boundaries. Time is included only as temperature is conducted and convected and as the velocity field is perturbed by the changing boundary. The current work investigates the viscoelastic effects with the simplifying assumptions of two dimensional, steady state flow.

To this end, this report contains a brief review of the finite element method, a discussion of the viscoelastic constitutive models used in the finite element equations, the details of the numerical schemes used in solving the equations, the computer implementation of the numerical schemes, a discussion of calculations conducted for four flow geometries to assess the numerical model, and an evaluation of the application of the numerical technique to the gyroscope fabrication.

II. THE FINITE ELEMENT METHOD IN FLUID DYNAMICS

This section is not intended to be exhaustive in nature, but rather to review some of the more important features of the finite element method employed in this work. References may be consulted for a more thorough treatment of the methods.

We begin by repeating that the finite element method is an approximate method of solving the differential equations of boundary and initial value problems (1,2). Field variables are solved by dividing the bounded region into subsets (finite elements) which themselves are governed by the differential equations. By approximating the distribution of the variables within each finite element by a trial function, the variables at any point in the element can be determined by a linear combination of the variable at specified points on the element edges. These points are called the nodes of the element; the variables at the nodes being determined by solving linear algebraic equations formed by assembling all of the elements into a matrix equation of order pqm, where p is the number of elements, q is the number of nodes per element, and m is the number of variables per node. The coefficients of the variables in the simultaneous equations are the integrals of the governing differential equation taken over the region of the element which is bounded by that node.

Mathematically, we write the discretization as:

$$\int_{\Omega} F d\Omega = \sum_{i=1}^{n} \int_{\gamma_i} F d\gamma_i = 0 \qquad \text{(II.1)}$$

with prescribed boundary conditions. In equation II.1, F is the governing differential equation, Ω is the entire region and γ_i is the region of the finite element. Where physical relations apply (such as the virtual work principle in solid mechanics), the equations can be formed in that basis. This is the approach used in references 1 and 9.

When the differential equation is self-adjoint (can be written in the form $(py')' + qy + f = 0$) with appropriate boundary conditions the equations can be formed by an abbreviated variational principle by merely multiplying the differential equation by the variation of the independent variables, i.e.

$$\int_{\gamma_i} [(py')' + qy + f]\delta y d\gamma_i = \delta I = 0 \qquad \text{(II.2)}$$

where I is the integral of the variational problem formed from the governing differential equation. Of course, this is merely stating that the euler equation of the variational principle is identical to the governing differential equation (see [13]). When the equations are not self adjoint, or the boundary conditions are unsuitable, an extremum principle can still be found, unless odd number derivatives are present. In that case, which is the situation with the complete Navier-Stokes equation with convection, a true extremum principle does not exist [14]. Formation of the finite element equations by a variational principle is the Ritz method. This method is most useful for the "creeping" flow solution of viscous fluids where the governing differential equation is known to be the euler equation of the proper extremum principle [15].

In the case of the complete Navier-Stokes equation, the method of weighted residuals is used wherein the error which remains after substituting appropriate trial functions into the governing equation is orthogonally projected to a set of weighting functions [2]. By setting the inner product of the error and the weighting function equal to zero, the approximate differential equation is then satisfied. Zienkiewicz [1] describes the two most popular methods of selecting the weighting functions as the Galerkin and Collocation methods. Due to its generality, the Galerkin method is the most popular for formulating the finite element equations for fluid flow problems. Selecting this method then the element variables are approximated by

$$a = \sum_{j=1}^{m} N_j C_j \tag{II.3}$$

where a is the field variable in the element, C_j are the values of the variable at the node points and N_j are the set of trial (basis) functions which satisfy the element boundary condition. When equation II.3 is substituted into the functional F of equation II.1, we obtain in general:

$$\sum_{i=1}^{n} \int_{\gamma_i} F(a)\,d\gamma_i = \sum_{i=1}^{n} \int_{\gamma_i} \varepsilon\,d\gamma_i \neq 0 \tag{II.4}$$

where ε is the residual error of the differential equation. Now using the Galerkin method of forming the inner product of the error and the trial functions we obtain:

$$\sum_{i=1}^{n} \int_{\gamma_i} N_k F(\sum_{j=1}^{m} N_j C_j) d\gamma_i = 0 \quad (k=1,m) \tag{II.5}$$

In this manner, we form m times n equations for the determination of the value of variable a at the points C_j.

In selecting the field variables to be approximated Frecaut [16] provides an excellent review of the advantages and disadvantages of the different formulations. The governing equations in an eulerian reference frame are continuity

$$\nabla \cdot \underline{u} = 0 \tag{II.6}$$

and momentum

$$\rho[\underline{u},t + (\underline{u}\cdot\underline{\nabla})\underline{u}] = b_o - \nabla p + \underline{\nabla}\cdot\underline{\sigma} \tag{II.7}$$

where in rectilinear flow:

$\underline{u}$ is the velocity vector $\begin{bmatrix} u \\ v \\ w \end{bmatrix}$

$\underline{b_o}$ is the body force vector $\begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$

$\underline{\sigma}$ is the deviatoric stress vector $\begin{bmatrix} \sigma_{11}\underline{i} + \sigma_{12}\underline{j} + \sigma_{13}\underline{k} \\ \sigma_{21}\underline{i} + \sigma_{22}\underline{j} + \sigma_{23}\underline{k} \\ \sigma_{31}\underline{i} + \sigma_{32}\underline{j} + \sigma_{33}\underline{k} \end{bmatrix}$

ρ is the constant density

p is the hydrostatic pressure

∇ is the gradient operator $\frac{\partial}{\partial x}\underline{i} + \frac{\partial}{\partial y}\underline{j} + \frac{\partial}{\partial z}\underline{k}$

and the comma denotes differentiation with respect to time. If the flow is purely viscous, the deviatoric stresses can be written as explicit functions of the velocity gradients leaving only velocity and pressure as independent variables. If both are approximated by the Galerkin method, the number of unknowns is relatively high (i.e. components of velocity at each node plus the pressure). In addition, some of the diagonal terms of the coefficient matrix become zero which limits the pivoting techniques generally used for solving the equations.

Two methods have been devised for eliminating the pressure. For two dimensional flow, the stream function $u = \psi_{,y}$ and $v = -\psi_{,x}$ is used to satisfy continuity and results in the disappearance of the pressure term when inserted into the momentum equation. However, the application to mixed boundary value problems is difficult, as shown by Tanner [17]. For incompressible problems, the penalty function formulation has been developed. This method, reviewed in detail in [7], replaces the incompressible continuity equation by the approximation

$$p = -\alpha\ (\nabla \cdot \underline{u}) \tag{II.8}$$

where α is a large positive number whose effect is to "penalize" the error of not satisfying continuity. In reference [7], it is shown that this method converges to the exact solution for "creeping" flow and that the selection of α is determined from the relation:

$$\alpha = c\mu \qquad \text{(II.9)}$$

Where c is a constant equal to 10^7 and μ is the dynamic viscosity. Furthermore, to avoid the trivial solution of $\underline{u} \to 0$ as $\alpha \to \infty$ (see equation II.10) the coefficients determined from evaluation of the integral must be singular. This is accomplished by employing reduced integration (quadrature rule of lower order than the exact for a given element) for the pressure term. The other terms can then be integrated at the optimum order (selective reduced integration) or at the lower order (uniform reduced integration). While it is more accurate to employ selective reduced integration (SRI), it is usually more convenient to use uniform reduced integration (URI) in the computer programs. Since it has been shown that 8 node quadrilateral elements exhibit inferior behavior to 9 node elements even for SRI, it is strongly recommended that when URI is used the 9 node "Lagrange" isoparametric element be employed [7].

Bercovier [18] has recently concluded that the reduced integration approach is only valid for straight-sided elements (biquadratic) if the governing equation is linear ("creeping" flow) and valid only for rectangular elements (vice bilinear quadrilaterals) when the equation is non-linear (with convection). Since most of our work concerns linear systems, this is not viewed as a limitation.

For ease of implementation, economy, and accuracy, therefore, we selected the penalty method with URI, 9 node Lagrange isoparametric elements. For comparison, some eight node "serendipity" element cases were run and will be discussed in Section VI.

Applying the Galerkin formulation of the finite element equations we obtain the following for two dimensional, rectilinear, incompressible, viscous flow:

$$(\underline{\underline{K}} + \overline{\underline{\underline{K}}} + \overline{\overline{\underline{\underline{K}}}})\ \hat{\underline{u}} + \underline{\underline{M}}\ \frac{\partial}{\partial t}\ \hat{\underline{u}} + \underline{f} = 0 \tag{II.10}$$

Where

$\hat{\underline{u}}$ is a column vector of the two dimensional velocities at the node points,

$\underline{\underline{N}}$ is the matrix of trial (shape) functions,

$$\underline{\underline{K}} = \int_\gamma \underline{\underline{B}}^T\ \underline{\underline{D}}\ \underline{\underline{B}}\ d\gamma$$

$$\overline{\underline{\underline{K}}} = \int_\gamma \rho\ \underline{\underline{N}}^T\ (\nabla \cdot (\underline{\underline{N}}\ u)^T)^T \underline{\underline{N}}\ d\gamma$$

$$\overline{\overline{\underline{\underline{K}}}} = \int_\gamma (\underline{m}^T\ \underline{\underline{B}})^T \alpha\, \underline{m}^T\ \underline{\underline{B}}\ d\gamma$$

$$\underline{\underline{M}} = \int_\gamma \underline{\underline{N}}^T \rho\ \underline{\underline{N}}\ d\gamma$$

and

$$\underline{f} = -\int_\gamma \underline{\underline{N}}^T\ \underline{b_o}\ d\gamma - \int_\Gamma \underline{\underline{N}}^T\ \underline{t}\ d\Gamma$$

In the matrix definitions above, we used the further identities:

$$\underline{\underline{D}} = \mu \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \underline{\underline{B}} = \underline{\underline{L}}\ \underline{\underline{N}} \text{ and } \underline{m} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \text{ where}$$

$\underline{\underline{L}}$ is the differential operator matrix for two dimensional flow

$$\underline{\underline{L}} = \begin{bmatrix} \frac{\partial}{\partial x} & o \\ o & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}.$$

Also the second term in the expression for $\underline{f}$ is the surface traction on the line element Γ which results from integrating the viscous stress term by parts. (Throughout this report a single underline denotes a vector quantity, and a double underline denotes a matrix quantity.)

When the inertial effects are comparable to the viscous ones, i.e., Reynolds No. greater than one, equations II.10 are non linear and must be solved by some iterative scheme. A discussion of these techniques will be postponed until the non-linear viscoelastic effects are added in Section IV.

Of course equation II.10 is the well known "weak" form of the Navier-Stokes governing differential equation which has been derived elsewhere by the virtual work statement [1].

III. VISCOELASTIC CONSTITUTIVE MODELS

The selection of a viscoelastic constitutive model (the rheological equation of state) for use in the finite element equations is generally a compromise between the accuracy of the model and ease of implementation. Because all of the models are nonlinear consideration must be given to the relative effects on the numerical convergence of the solutions. In this study, two general ground rules were used in selecting the appropriate model. First, for the material under consideration (fiber-filled polyphenylene sulfide), adequate rheological or viscometric data do not exist to justify the use of multiple constant models, and second only a first order effect on the flow field was being sought. Once success is achieved in modeling viscoelasticity, rheological data can be obtained and adjustments to the constitutive model investigated.

As before, only essential elements for understanding the behavior of the selected viscoelastic model are presented in this report. For a thorough discussion of the continuum mechanics of viscoelastic materials the references can be consulted (19, 20, 21, 22, 23).

For a fluid element, the resistance to deformation when a force is applied can be thought of as a combination of viscous and elastic stresses. Modeling these as a dashpot and spring respectively as shown in Figure 3, we obtain the well-known Maxwell Element for fluids. Using the nomenclature of Figure 3,

where μ is the dynamic viscosity, G is the shear modulus of elasticity, ε is the infinitesimal strain and σ is the applied shear stress we obtain the stress-strain rate relation:

$$\dot{\varepsilon} = \frac{\dot{\sigma}}{G} + \frac{\sigma}{\mu} \tag{III.1}$$

Generalizing to a three-dimensional form, we have:

$$\underline{\underline{\sigma}} + \lambda \frac{\partial}{\partial t} (\underline{\underline{\sigma}}) = 2\mu \underline{\underline{d}} \tag{III.2}$$

where $\underline{\underline{\sigma}}$ is the Cauchy deviatoric stress tensor

μ is the dynamic viscosity

$\lambda = \mu/G$ is a time constant known as the relaxation time

and $\underline{\underline{d}}$ is the rate of deformation tensor whose components are defined as:

$$d_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad (i,j = 1,2,3) \tag{III.3}$$

Equation III.2 is suitable when the rate of deformation in the flow is infinitesimually small. But for general motion, in which the rate of deformation is not necessarily small, the time derivative of the Cauchy stress tensor violates two fundamental requirements of any equation of state. These requirements are that the equation describes material properties independent of the frame of reference, and that the behavior of any material element must depend only on its previous deformative history and not in any way on the state of neighboring elements, or on rigid body translation/rotation. These discrepancies are corrected by substituting for the time derivative of the Cauchy stress either an Oldroyd derivative [8] (known as a convected or a codeformational derivative) or a Jaumann derivative [24] (known as a

co-rotational derivative). These modifications will be discussed shortly. Once the above requirements are satisfied it only remains to tailor the equation so as to fit experimental observations. This is done by introducing added parameters which are multiplied by functions of the invariants of the rate of strain tensor.

Han [23] presents a survey of the major refinements developed for the two invariant stress derivatives along with the material properties they predict. A two constant (λ, μ) model using an Oldroyd derivative is known as a White-Metzner model. When the Jaumann derivative is used, the equation is called a DeWitt model. As multiple parameters are added, the general models are known merely as n-order Oldroyd models. Two other models derived by means somewhat different from the generalized Maxwell element are the Spriggs model which builds many Maxwell elements at the molecular structure level and the Rivlin Erickson fluid which merely states that the fluid stress is a function of the invariants of the gradients of displacement, velocity, acceleration, second acceleration, and so on.

Returning to the invariant stress derivatives, we write them explicitly for further discussion. For the Oldroyd derivative in contravariant form (see [22] for a discussion of covariant and contravariant tensors) we obtain:

$$\frac{\delta \sigma_{ij}}{\delta t} = \frac{\partial \sigma_{ij}}{\partial t} + u_k \frac{\partial \sigma_{ij}}{\partial x_k} - \sigma_{kj} \frac{\partial u_i}{\partial x_k} - \sigma_{ik} \frac{\partial u_j}{\partial x_k} \qquad \text{(III.4)}$$

Where the range and summation indicial convention is used.

Similarly the Jaumann derivative is:

$$\frac{\mathcal{D}\sigma_{ij}}{\mathcal{D}t} = \frac{\partial\sigma_{ij}}{\partial t} + u_k \frac{\partial\sigma_{ij}}{\partial x_k} + \omega_{ik}\,\sigma_{ik} + \omega_{jk}\,\sigma_{ik} \qquad \text{(III.5)}$$

where $\omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)$ are the components of the vorticity tensor. Again, see Han [23] for an excellent discussion of the physical significance of the terms on the right-hand side of equations III.4 and III.5.

We will also have occasion to discuss further the Rivlin-Ericksen fluid so we list the general equation for an incompressible fluid:

$$\underline{\underline{\sigma}} = \alpha_1 A_{(1)} + \alpha_2 A^2_{(1)} + \alpha_3 A_{(2)} + \alpha_4 A^2_{(2)} + \alpha_5 (A_{(1)}A_{(2)} + A_{(2)}A_{(1)}$$
$$+ \alpha_6 (A^2_{(1)}A_{(2)} + A_{(2)}A^2_{(1)}) + \alpha_7 (A^2_{(2)}A_{(1)} + A_{(1)}A^2_{(2)})$$
$$+ \alpha_8 (A^2_{(1)}A^2_{(2)} + A^2_{(2)} + A^2_{(2)}A^2_{(1)}) \qquad \text{(III.6)}$$

where the α_i are functions of the invariants of $A_{(1)}$ and $A_{(2)}$ and

$$A^{(1)}_{ij} = 2d_{ij}$$

$$A^{(2)}_{ij} = \frac{\partial A^{(1)}_{ij}}{\partial t} + u_k \frac{\partial A^{(1)}_{ij}}{\partial x_k} + A^{(1)}_{kj}\frac{\partial u_k}{\partial x_i} + A^{(1)}_{ik}\frac{\partial u_k}{\partial x_j}$$

In passing, it is noted that the preceding discussion of models has focused on the rate type. If equation III.2 is integrated with respect to time rheological equations of state of the integral type are obtained. While this type proves useful for some rheological investigation, it complicates finite element calculations by requiring a complete time history of the strain path of all elements.

Finally, before we can discuss the relative merits of the models, we must make some definitions. Steady simple shear flow, also known as viscometric flow, is defined by the velocity field

$$u = \dot{\gamma}y,\ v = w = o \tag{III.7}$$

where $\dot{\gamma}$ is a constant shear strain rate and

u is the velocity normal to the y axis of the cartesian coordinate system.

Substituting equation III.7 into III.3 we find the rate of deformation tensor to be:

$$\underline{\underline{d}} = \begin{bmatrix} o & \dot{\gamma}/2 & o \\ \dot{\gamma}/2 & o & o \\ o & o & o \end{bmatrix} \tag{III.8}$$

For viscometric flow, viscoelastic fluids are observed to exhibit three independent material properties, the standard viscosity, and a first and second normal stress function written consecutively as:

$$\sigma_{12} = \mu(\dot{\gamma})\dot{\gamma},\sigma_{11} - \sigma_{22} = \psi_1(\dot{\gamma})\dot{\gamma}^2,\ \sigma_{22} - \sigma_{33} = \psi_2(\dot{\gamma})\dot{\gamma}^2 \tag{III.9}$$

Implicit in equations III.9 is the further observation that when a fluid behaves viscoelastically, the material parameters are not constant, but vary with the rate of strain. This non-newtonian behavior is generally observed to follow a power law relation, written for the viscosity as:

$$\mu(\dot{\gamma}) = \frac{\mu_o}{1 + (K/\mu_o)^{-1} (\dot{\gamma}/\ 2)^{(1-n)}} \qquad \text{(III.10)}$$

where μ_o, K, and n are parameters selected emperically. When the exponent n is less than zero, the viscosity varies inversely to the shear strain rate and the fluid is termed shear-thinning. When n is greater than zero the fluid is shear thickening. Most real fluids are shear thinning. ψ_1 and ψ_2 on the other hand are observed to increase exponentially with shear strain rate.

Before we continue, recall that equation III.10 was written for simple shear flow. This equation is merely the specialization of the more commonly written general flow form:

$$\mu(II_d) = \frac{\mu_o}{1 + (K/\mu_o)^{-1} (\tfrac{1}{2}\ II_d)^{(1-n)/2}} \qquad \text{(III.11)}$$

where II_d is the second invariant of the rate of strain tensor

$$II_d = d_{ij}d_{ij}, \qquad \text{(III.12)}$$

which in two dimensional rectilinear flow can be written explicitly as :

$$II_d = 4\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right] + 2\left[\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right]^2 \qquad \text{(III.13)}$$

We are now prepared to make a selection of the constitutive equation to implement in the finite element equations. The choices have been narrowed to (i) White-Metzner (ii) DeWitt and (iii) Rivlin-Ericksen as generally representative of the available models (Pipkin and Tanner [25] present a thorough review of all the models for viscosmetric flow). Middleman [26] has

presented an excellent discussion of the correlation of the properties predicted by the White-Metzner and DeWitt models to experimental observations. In simple shear flow, the DeWitt model is somewhat superior because the second normal stress function is finite whereas the White-Metzner model predicts that it vanishes. However, in general flow fields the DeWitt model varies appreciably from reality while the White-Metzner model maintains consistency. Since ψ_2 is generally small, the fact that the White-Metzner model predicts a zero value is not considered a major drawback by Middleman and we agree. Han [23] suggests that since the Oldroyd derivative takes a different form if written in terms of covariant or contravaraint basis vectors that it is inferior to the Jaumann derivative. Since the work herein is conducted for a rectilinear coordinate system, it is felt that this is less of a penalty than the cited deviation of the Jaumann derivative model for general flow fields. Therefore, the author concurs with Middleman's recommendation that the White-Metzner model is preferred to the DeWitt model.

Considering the Rivlin-Ericksen fluid, Tanner [17] notes that for a simple shear flow equation III.6 reduces to:

$$\sigma_{ij} = \mu A_{ij}^{(1)} + (\psi_1 + \psi_2) A_{ik}^{(1)} A_{kj}^{(1)} - \tfrac{1}{2}\psi_1 \; A_{ij}^{(2)} \qquad \text{(III.11)}$$

Clearly equation III.6 is overly complicated for our initial work. But since the simplification to III.11 presumes simple shear flow, it is disqualified as a candidate for this effort. It is interesting to note, however, that of the three models

considered, the Rivlin Ericksen fluid alone permits the deviatoric stress to be written as an explicit function of the velocities and nth order derivatives of velocities. The advantages of this fact will become obvious in the next section when we discuss the formation and solution of the complete finite element equation.

Let us recapitulate before concluding this section. A White-Metzner modified Maxwell element was selected for the rheological equation of state because of its ability to approximate real viscoelastic fluid behavior while requiring only two model parameters. In addition, the two parameters μ and λ are taken to be functions of the second invariant of the rate of strain tensor as defined in equation III.11.

For plane, steady flow where $w = \frac{\partial}{\partial z} = \frac{\partial}{\partial t} = o$, the nine equations of III.2 reduce to four which are written explicitly below with the use of equation III.4.

$$\sigma_{xx} + \lambda\left(u \frac{\partial\sigma_{xx}}{\partial x} + v \frac{\partial\sigma_{xx}}{\partial y} - 2\sigma_{xx} \frac{\partial u}{\partial x} - \sigma_{yx} \frac{\partial u}{\partial y}\right) = 2\mu \frac{\partial u}{\partial x} \qquad \text{(III.12a)}$$

$$\sigma_{xy} + \lambda\left(u \frac{\partial\sigma_{xy}}{\partial x} + v \frac{\partial\sigma_{xy}}{\partial y} - \sigma_{xx} \frac{\partial u}{\partial y} - \sigma_{xy} \frac{\partial u}{\partial x} - \sigma_{xy} \frac{\partial v}{\partial y}\right) = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \qquad \text{(III.12b)}$$

$$\sigma_{yx} + \lambda\left(u \frac{\partial\sigma_{yx}}{\partial x} + v \frac{\partial\sigma_{yx}}{\partial y} - \sigma_{xx} \frac{\partial v}{\partial x} - \sigma_{yy} \frac{\partial u}{\partial y} - \sigma_{yx} \frac{\partial u}{\partial x} - \sigma_{yx} \frac{\partial v}{\partial y}\right) = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \qquad \text{(III.12c)}$$

$$\sigma_{yy} = \lambda\left(u \frac{\partial\sigma_{yy}}{\partial x} + v \frac{\partial\sigma_{yy}}{\partial y} - 2\sigma_{yy} \frac{\partial v}{\partial y} - \sigma_{xy} \frac{\partial v}{\partial x} - \sigma_{yx} \frac{\partial v}{\partial x}\right) = 2\mu \frac{\partial v}{\partial y} \qquad \text{(III.12d)}$$

These equations are identical to those used by Perera and Strauss [27] in their finite difference formulation of similar problems when account is made of the reduction of the four-constant model they used vice the two parameter model used herein.

The reader is reminded that the stresses in equations III.12 are the deviatoric ones and differ from the complete stresses by the hydrostatic pressure. Since the momentum equation always expresses these two stresses separately, they are not combined here either.

IV. VISCOELASTIC FINITE ELEMENT EQUATIONS

The governing differential equations for an incompressible viscoelastic fluid are as presented in equations II.6 and II.7. Continuity and momentum are repeated:

$$\nabla \cdot \underline{u} = o \qquad \text{(IV.1a)}$$

$$\rho \, [\underline{u},_t + (\underline{u} \cdot \nabla)\underline{u}] = \underline{b}_o - \nabla p + \nabla \cdot \underline{\sigma} \qquad \text{(IV.1b)}$$

The boundary conditions of course will be for the independent variables and gradients of these variables. However, for many flow problems, it is more convenient to specify the tractions (stresses) on some boundaries and the independent variables on others. This is the mixed boundary condition formulation and is of course mandatory for finite element equations which are reduced to a set of inhomogeneous linear algebraic equations. While specification of the variables ($\underline{u}$, ψ, p, $\underline{\sigma}$ depending on the type of equations used) at the boundaries is straight forward, the specification of boundary tractions must be consistent with the type of problem. For example, Chang [15] discusses the difference in specifying the surface traction, for a number of flow cases, between a non-newtonian viscous fluid and a generally viscoelastic one. Understanding these differences is particularly important when a specific type of flow is prescribed (e.g., fully developed entry flow) for an assessment of the accuracy of the finite element model. We defer further comment on the boundary conditions until Section VI when specific flow problems are considered.

Briefly reviewing the past work on finite element modeling of viscoelastic flow, it is noted that no investigations, known to the author, have been conducted using the "penalty" method for incompressible fluid flow. Tanner [17] and Caswell and Tanner [12] have used the formulation with velocities and pressure as the independent variables, with a Rivlin-Ericksen fluid for viscometric flow. Results have been excellent for power law fluids, but only Poiseuille flow has been considered for the viscoelastic case. Kawahara and Takeuchi [28] applied a mixed method where the total deviatoric stress (viscous and elastic) was independently interpolated along with the velocities and pressure. The White-Metzner constitutive equation was then solved simultaneously with the Navier-Stokes equation for incompressible fluids. Using six-node triangular elements in plane flow, this gives rise to eighteen additional unknowns per element and is felt to have limitations for general problems because of the computer capacity required for large, complicated geometric problems. However, they did achieve good results for expanding and bending flow through channels for relaxation times up to 0.1 seconds.

In the work most similar to the current effort, Chang et. al. [15] solved the equations using the White-Metzner model with velocities and pressure the field variables for the finite element equations. In two-dimensional, steady state, convective, isothermal flow, the slip stick problem was solved for material cases of Weissenberg numbers up to 0.2.

The Weissenberg number is a dimensionless ratio of recoverable or elastic shear stress to total shear stress in steady flow. It is written

$$Ws = \frac{\lambda U}{L} \qquad \text{(IV.1)}$$

where λ is the relaxation time in seconds,

U is a characteristic steady velocity in cm/sec,

and L is a characteristic length in cm.

Han [23] presents rheological data for high and low density polyethylene at various shear strain rates (U/L) at 200^{o}C. For high density polyethylene, the Ws varies from 35 at 0.025 cm/cm-sec down to 0.01 at 100 cm/cm-sec. On the other hand, the Ws for low density polyethylene varies between 5 at the low strain rate and 0.01 at the high strain rate. We note that this is essentially the range of interest for practical problems ($0.01 \leq Ws \leq 35$). A major difficulty in the finite element method has been obtaining numerical convergence for problems of relatively high Ws as evidenced in the above review. It appears that Chang's work has provided the highest value. Without discussion, it is noted that with this convergence problem, the added numerical problems associated with evaluation of the pressure term in the tangent stiffness matrix for the penalty method may suggest some limitations in the future for application to viscoelasticity.

Now using the Galerkin formulation with the penalty method, equations IV.1 become for steady state

$$\left\{ \int_V (\underline{N}^T \rho (\nabla \cdot (\underline{N}\ \hat{\underline{u}})^T)^T \underline{N} + (\underline{m}^T\ \underline{B})^T \alpha \underline{m}^T\ \underline{B}) dv \right\} \hat{\underline{u}} - \int_V \underline{N}^T \underline{L}^T \underline{\sigma} dv = o \qquad \text{(IV.2)}$$

Where all terms have been defined in equation II.10, the body forces are assumed to be zero, and two-dimensional rectilinear flow is treated so that the plane stress vector $\underline{\sigma}$ is:

$$\underline{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} \tag{IV.3}$$

We now split the deviatoric stess into a viscous and elastic portion

$$\underline{\sigma} = \underline{\sigma}^{v} + \underline{\sigma}^{e} \tag{IV.4}$$

substitute into equation IV.2, and apply Green's divergence theorem to obtain

$$\left\{ \int_{v} (\underline{\underline{B}}^{T} \underline{\underline{D}}\ \underline{\underline{B}} + \underline{\underline{N}}^{T} \rho (\nabla \cdot \underline{\underline{N}}\hat{\underline{u}})^{T})^{T} \underline{\underline{N}} + (\underline{m}^{T} \underline{\underline{B}})^{T} \alpha \underline{m}^{T} \underline{\underline{B}})\, dv \right\} \hat{\underline{u}} + \int_{v} \underline{\underline{N}}^{T} \underline{\underline{L}}^{T} \underline{\sigma}^{e} dv - \int_{A} \underline{\underline{N}}^{T} \underline{t}\, dA = o \tag{IV.5}$$

where the viscous stress has been written explicity as

$$\underline{\sigma} = \underline{\underline{D}}\ \underline{\underline{L}}\ \underline{\underline{N}}\ \hat{\underline{u}} = \underline{\underline{D}}\ \underline{\underline{B}}\ \hat{\underline{u}} \tag{IV.6}$$

and the last term is the traction on the boundary. From equation III.2 we can write

$$(\underline{\sigma}^{v} + \underline{\sigma}^{e}) + \lambda \frac{\delta \underline{\sigma}}{\delta t} = 2\mu \dot{\underline{\varepsilon}} \tag{IV.7}$$

or since $\underline{\sigma}^{v} = 2\mu\dot{\underline{\varepsilon}}$

$$\underline{\sigma}^{e} = -\lambda \frac{\delta \underline{\sigma}}{\delta t} \tag{IV.8}$$

where $\dot{\underline{\varepsilon}}$ is the 2D rate of deformation vector.

From equations III.12, we see that for steady state equation IV.8 is of the following functional form

$$\underline{\sigma}^e = g(\underline{u}, \underline{\sigma}^e, \underline{\sigma}^v, \underline{\sigma}^{e\prime}, \underline{\sigma}^{v\prime}, \underline{u}', x, y) \qquad \text{(IV.9)}$$

Where the prime denotes differentiation with respect to x and y. But since $\underline{\sigma}^v$ is a unique function of $\underline{u}'$ we can further state

$$\underline{\sigma}^e = h(\underline{u}, \underline{u}', \underline{u}'', x, y, \underline{\sigma}^e, \underline{\sigma}^{e\prime}). \qquad \text{(IV.10)}$$

Equation IV.10 now makes equation IV.5 not only non-linear (even for creeping flow), but inexpressible in an explicit form. The equation must, therefore, be solved simultaneously with equation IV.5. This is the same point reached by Chang [15] and Perera [27]. Let us examine the method of solution proposed in [15]. Although convection was included in that analysis, it is easier to consider creeping flow (without loss of generality).

The creeping, viscoelastic flow can be written as:

$$\underline{\underline{K}}\,\hat{\underline{u}} + \underline{K}^e(\hat{\underline{u}}, \hat{\underline{u}}', \hat{\underline{u}}'', \underline{\sigma}^e, \underline{\sigma}^{e\prime}) = \underline{f} \qquad \text{(IV.11)}$$

where the terms $\underline{K}^e$ are the functional form of the internal elastic forces. Newton-Raphson iteration can not be employed to solve IV.11 because of the implicit dependent variable $\underline{\sigma}^e$. Instead the common method is to use successive substitution where an initial value of $\underline{\sigma}^e$ is guessed and substituted into equation IV.10. Assuming $\hat{\underline{u}}$ has first been solved for the linear problem, $\underline{K}^e$ can now be calculated, substituted into equation IV.11 and a new value of $\hat{\underline{u}}$ found. This new value of $\hat{\underline{u}}$ is than used with the latest value of $\underline{\sigma}^e$ to calculate an updated value of $\underline{\sigma}^e$ and the process is repeated until some convergence criterion is

satisfied. In terms of a solution for $\hat{\underline{u}}$ at iteration s+1, we have:

$$\underline{\underline{K}}\ \hat{\underline{u}}^{s+1} + \underline{K}^{e^s} = \underline{f}$$

and $\underline{\sigma}^{e^s} = h\left(\hat{\underline{u}}^s, \hat{\underline{u}}^{,s}, \hat{\underline{u}}^{,,s}, x, y, \underline{\sigma}^{e^{s-1}}, \underline{\sigma}^{e^{,s-1}}\right)$. (IV.12)

The actual calculation on the computer was performed at the iteration s+1 by subtracting $\underline{K}^{e^s}$ from $\underline{f}$ and solving $\underline{\underline{K}}\ \hat{\underline{u}}^{s+1}$. Therefore, the computer equation is:

$$\underline{\underline{K}}\ \Delta\ \hat{\underline{u}} = \underline{f} - \underline{K}^{e^s} - \underline{\underline{K}}\ \hat{\underline{u}}^s$$

where $\Delta\hat{\underline{u}} = \hat{\underline{u}}^{s+1} - \hat{\underline{u}}^s$. If the convection non-linearity is included the Picard substitution can be nested within a Newton-Raphson iteration.

If we momentarily disregard the issue of convergence, the only problem which remains is the calculation of the elastic stress gradient at the s-1 iteration. Chang [15] is completely silent on this issue and it is felt that it was ignored. Later on, we will discuss possible situations where this might be valid. To aid in the discussion, let us write equation III.12 in vector form by recognizing $\sigma_{xy} = \sigma_{yx}$. It can be verified that the equation becomes:

$$\underline{\sigma}^e = \lambda[\underline{\underline{A}}\ \underline{\sigma} - (\underline{u} \cdot \nabla)\underline{\sigma}] \qquad \text{(IV.13)}$$

Where $$\underline{\underline{A}} = \begin{bmatrix} 2\frac{\partial u}{\partial x} & o & 2\frac{\partial u}{\partial y} \\ o & 2\frac{\partial v}{\partial y} & 2\frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial x} & \frac{\partial u}{\partial y} & \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \end{bmatrix}$$

and
$$(\underline{u} \cdot \nabla)\underline{\sigma} = u \frac{\partial \underline{\sigma}}{\partial x} + v \frac{\partial \underline{\sigma}}{\partial y}$$

For one-dimensional flow equation IV.13 becomes

$$\sigma^e_{xx} = a\sigma^e_{xx} - u \frac{\partial \sigma^e_{xx}}{\partial x} + b \qquad \text{(IV.14)}$$

where a and b are the appropriate functions of u and $\frac{\partial u}{\partial x}$. It is convenient to use this equation to discuss the methods of solution for the first order non-linear differential equation.

Equation IV.14 is the identical form of the Picard method of first order eouations namely [29]: $\frac{dy}{dx} = F(x,y)$ where σ^e_{xx} corresponds to y and a, u, and b are functions only of x. The equation is integrated yielding

$$y = y_o + \int_{x_o}^{x} F(x,y)\,dx$$

where y_o is the initial value at x_o.

Equation IV.14 would become:

$$\sigma^e_{xx} = \sigma^e_{xx_o} + \int_{x_o}^{x} \frac{1}{u}\left[(a-1)\sigma^e_{xx} + b\right] dx \qquad \text{(IV.15)}$$

Assuming the integral could be evaluated numerically σ^e_{xx} could be solved by the same successive substitution scheme used for the complete finite element equations. An initial guess is made for σ^e_{xx} in the integrand and the right-hand side is solved for an updated value of σ^e_{xx}. That value is then substituted into the integrand and the procedure repeated until convergence is achieved. Let us now write IV.13 in this form

$$(\underline{u}\cdot\nabla)\underline{\sigma} = \frac{1}{\lambda}\,(\underline{\sigma}^e - \underline{\underline{A}}\,\underline{\sigma}), \qquad \text{(IV.16)}$$

and upon integration by taking the dot product of both sides with $dA = dx\underline{i} + dy\underline{j}$

$$(u + v)\underline{\sigma} = \underline{\sigma}_o + \int_{A_o}^{A} \frac{1}{\lambda}\left(\underline{\sigma}^e - \underline{\underline{A}}\,\underline{\sigma}\right)\cdot dA \qquad \text{(IV.17)}$$

While in theory, IV.17 could be solved, it is felt that in a finite element formulation, it would be impractical to use such a system that requires an initial value to be calculated at a corner of each element ($\underline{\sigma}_o$) and separate integration of the spatial derivatives, i.e.,

$$\int_{A_o}^{A} \frac{1}{\lambda}\left(\underline{\sigma}^e - \underline{\underline{A}}\,\underline{\sigma}\right)\cdot dA = \int_{x_o}^{x} \frac{1}{\lambda}\left(\underline{\sigma}^e - \underline{\underline{A}}\,\underline{\sigma}\right)dx + \int_{y_o}^{y} \frac{1}{\lambda}\left(\underline{\sigma}^e - \underline{\underline{A}}\,\underline{\sigma}\right)dy \neq \int_{A_o}^{A} \frac{1}{\lambda}\left(\underline{\sigma}^e - \underline{\underline{A}}\,\underline{\sigma}\right)dx\,dy$$

Due to the difficulties encountered, another method was sought for the solution of IV.13. If the derivative is approximated by a Taylor series, then a standard finite difference equation is achieved and usual relaxation methods can be employed for the solution. Referring to Figure 4 and using central differences we have for the first component of $\underline{\sigma}^e$

$$
{}^{i,j}\sigma^{e}_{xx} = \lambda^{i,j}\Big\{ 2\,\frac{\partial u}{\partial x}^{i,j}\Big(2\mu^{i,j}\,\frac{\partial u}{\partial x}^{i,j} + {}^{i,j}\sigma^{e}_{xx}\Big) + 2\,\frac{\partial u}{\partial y}^{i,j}\Big(
\mu^{i,j}\Big[\frac{\partial v}{\partial x}^{i,j} + \frac{\partial u}{\partial y}^{i,j}\Big] + {}^{i,j}\sigma^{e}_{xy}\Big)
- u^{i,j}\mu^{i,j}\,\frac{\partial^2 u}{\partial x^2}^{i,j} - v^{i,j}\,\mu^{i,j}\,\frac{\partial^2 u}{\partial x \partial y}^{i,j} - u^{i,j}\,\frac{\partial \sigma^{e}_{xx}}{\partial x}\Big|_{i,j} - v^{i,j}\,\frac{\partial \sigma^{e}_{xx}}{\partial y}\Big|_{i,j}\Big\}
$$

(IV.18)

Where

$$
\frac{\partial \sigma}{\partial x}\Big|_{i,j} = \frac{\left(\sigma^{i+1,j}-\sigma^{i-1,j}\right)\left(y^{i,j+1}-y^{i,j-1}\right)-\left(\sigma^{i,j+1}-\sigma^{i,j-1}\right)\left(y^{i+1,j}-y^{i-1,j}\right)}{\left(x^{i+1,j}-x^{i-1,j}\right)\left(y^{i,j+1}-y^{i,j-1}\right)-\left(y^{i+1,j}-y^{i-1,j}\right)\left(x^{i,j+1}-x^{i,j-1}\right)}
$$

and

$$
\frac{\partial \sigma}{\partial y}\Big|_{i,j} = \frac{\left(\sigma^{i,j+1}-\sigma^{i,j-1}\right)\left(x^{i+1,j}-x^{i-1,j}\right)-\left(\sigma^{i+1,j}-\sigma^{i-1,j}\right)\left(x^{i,j+1}-x^{i,j-1}\right)}{\left(x^{i+1,j}-x^{i-1,j}\right)\left(y^{i,j+1}-y^{i,j-1}\right)-\left(y^{i+1,j}-y^{i-1,j}\right)\left(x^{i,j+1}-x^{i,j-1}\right)}
$$

(IV.19)

Equations IV.19 are derived in Appendix I.

A few words about equations IV.18 and IV.19 are in order. While central differences are expected to give higher order accuracy, Roache [30] notes that the numerical stability is much poorer than backward (upwind) differencing and for a non-uniform mesh (special mesh system), it is very likely that the approximation deteriorates from third order accuracy in the mesh point spacing to first order accuracy. In IV.18, the viscous stress is expressed in terms of the equivalent rate of strain through

equation IV.6. Also the expressions for the gradient of viscous stresses appear to treat the dynamic viscosity as independent of x and y. This is not the case. Rather it can be seen upon differentiation of the products $\frac{\partial}{\partial x}\left(\mu(x,y)\,\frac{\partial u}{\partial x}\right)$ for example, that for a power law fluid the term $\frac{\partial u}{\partial x}\left(\frac{\partial \mu}{\partial x}\right)$ is of higher order and, therefore, is neglected. Finally all terms in IV.19 are elastic xx stresses. The subscripts and superscripts have been dropped so as to not severely encumber the equations. Equation IV.18 is a first order derivative counterpart to the steady, convection-dissipation finite difference equation which gives rise to classic under and over-relaxation methods. However, we do not have an equivalent Courant number so we merely employ Richardson/ Jacobi iteration. Calling the left-hand side of IV.18 iteration k+1 and the elastic stress terms on the right-hand side iteration k (which is known) we sweep through the entire solution domain in the relaxation process. As in most cases, $\underline{\sigma}^e$ at the first iteration k=1 is assumed to be zero. The issues which we must discuss in solving IV.18 by this technique are the selection of mesh points i,j, evaluation of the second derivative of the velocity, convergence of the iteration, and treatment of boundary elements where boundary conditions must be invoked. We will take these in the listed order.

Since all the terms involving the field variable $\underline{u}$ in equation IV.18 are routinely calculated, in the evaluation of the integrals of the finite element equations, at the Gauss points in the Gauss quadrature it is natural to select these points as the mesh for the elastic stresses. Then for the

differences required in evaluating the elastic stress gradients, the elastic stress at Gauss points of adjacent elements can be used. This procedure is shown in Figure 4 for one of the Gauss points. Of course, two concerns arise. Procedurally, most finite element routines calculate element quantities such as velocity gradients in subroutines which dump the values upon exiting the subroutine, returning only values of global tangent stiffness components. Therefore, special schemes must be devised to identify, maintain, and pass current values of elastic stresses, external to the subject element, to the element for an update of the elastic stress at its Gauss points. Second, the discontinuity of stresses between elements which gives rise to the practice of "smoothing" must be recognized. At the early stages of iteration, this might aggravate the numerical stability. For this study, the first issue was resolved by programming techniques (principly by creating arrays which were stored in common memories between subroutines). The second issue was not addressed.

For the problem of the evaluation of the second derivative (recall from Section II we are using a "weak" form of the equations so that only C_o continuity is required of $\underline{u}$), we now require C_1 continuity of the trial functions and explicitly evaluate the term just as is done for the first derivative. To do this, a subroutine was written (ESHAP) which returns the values $\frac{\partial^2 N_i}{\partial x^2}$, $\frac{\partial^2 N_i}{\partial y^2}$, $\frac{\partial^2 N_i}{\partial x \partial y}$, at the Gauss points of an element.

The values $\frac{\partial^2 u}{\partial x^2}$, etc. are then calculated in the exact same manner as done for the first derivatives. For this subroutine of course, it was also necessary to calculate the determinant of the Jacobian of the second derivatives. The mathematics involved in subroutine ESHAP are given in Appendix 2.

Considering for the time being only convergence of the Richardson/Jacobi iteration scheme, (Newton-Raphson and Picard iteration are briefly treated later). We can apply the Lax/Richtmyer amplification matrix error method [31] discussed in [2]. Briefly, we write equation IV.13 in terms of the final value and errors at each iteration or

$$(\underline{\sigma}^e + \underline{\varepsilon})^{k+1} = \lambda \left[\underline{\underline{A}}(\underline{\sigma}^v + \underline{\sigma}^e + \underline{\varepsilon}) - (\underline{u}\cdot\nabla)(\underline{\sigma}^v + \underline{\sigma}^e + \underline{\varepsilon}) \right]^k \tag{IV.20}$$

Subtracting IV.13 from IV.20 we get

$$\underline{\varepsilon}^{k+1} = \lambda(\underline{\underline{A}} - (\underline{u}\cdot\nabla))\,\underline{\varepsilon}^k \tag{IV.21}$$

or

$$\frac{\underline{\varepsilon}^{k+1}}{\underline{\varepsilon}^k} = \lambda(\underline{\underline{A}} - \underline{u}\cdot\nabla) \leq 1 \tag{IV.22}$$

The test for convergence then is for the eigen values of the matrix $\lambda(\underline{\underline{A}} - u\cdot\nabla)$ to be ≤ 1. Note that the dimensions of this tridiagonal matrix are 3np where n is the number of Gauss points per element and p is the number of elements. The complete matrix is formed by assembling the individual 3x3 matrices at each Gauss point. We did not conduct any further analysis

of convergence, but rather have established bounds emperically. Little emphasis was placed on this issue because it was found during the course of the study that the outer iteration of equation IV.12 generally controlled convergence.

Finally at the boundary elements where an adjacent element may not exist, it is necessary to devise an auxiliary scheme for the calculation of $\frac{\partial \underline{\sigma}^e}{\partial x}$ and $\frac{\partial \underline{\sigma}^e}{\partial y}$ at the Gauss points. If $\underline{\sigma}^e$ is known at the element edges (in particular the node points) the nodes can be used as the forward or backward mesh points and the relaxation procedure continued. However, there are some major drawbacks to this. First regardless of the boundary condition (velocity or traction specified) additional calculations for velocity gradients and viscous stress gradients at the nodes must be accomplished. Additionally, the elastic stress gradient can not employ central differences at the node, but must be based on a backward difference. Third, the formation of the two independent equations to simultaneously solve $\frac{\partial \underline{\sigma}^e}{\partial x}$ and $\frac{\partial \underline{\sigma}^e}{\partial y}$ is quite cumbersome. A different technique was therefore developed.

A new common array was established (BOSIG) to identify and pass the elastic stresses at the four corner nodes. At the first iteration, these stresses (four nodes by three stress components by the number of elements) are initialized at zero. The velocity vector $\hat{\underline{u}}$ is then calculated in the Picard iteration. Then during the calculation of element values at the Gauss points (velocity, velocity gradient, stress gradients, etc.), the boundary elastic stress at the corner node which

matches the Gauss point is calculated according to:

$$\underline{\sigma}_e^{N.P.} = \underline{\sigma}_e^{G.P.} + \left.\frac{\partial \underline{\sigma}_e}{\partial x}\right|_{G.P.} (x^{N.P.} - x^{G.P.}) + \left.\frac{\partial \underline{\sigma}_e}{\partial y}\right|_{G.P.} (y^{N.P.} - y^{G.P.}) \tag{IV.23}$$

Where N.P. is the node point and G.P. is the Gauss point. This value of elastic stress is then used in the central difference calculation at the Gauss point if the element is on a boundary. Figure 5 shows the details for the calculation at the Gauss points for both boundary and interior elements as described above.

To keep our thoughts clear, it is instructive to pause and review. The creeping flow finite element equation to be solved is:

$$\left\{ \int_V (\underline{\underline{B}}^T \underline{\underline{D}}\, \underline{\underline{B}} + (\underline{m}^T \underline{\underline{B}})^T \alpha\, \underline{m}^T \underline{\underline{B}})\, dv \right\} \hat{\underline{u}} + \int_V \underline{\underline{N}}^T \underline{\underline{L}}^T \underline{\sigma}^e dv = \int_A \underline{\underline{N}}^T \underline{t}\, dA$$

The coefficients of $\hat{\underline{u}}$ are linear and $\underline{\sigma}^e$ is solved by successive substitution for each value of $\hat{\underline{u}}$. Notice two things. First, $\underline{\underline{N}}^T \underline{\underline{L}}^T = \underline{\underline{B}}^T$ so that we could make this substitution. This study, however, included the terms $\nabla \underline{\sigma}^e$ in the equation and so these values were used directly with $\underline{\underline{N}}^T$ in calculating the integral. Second, a nested iteration on $\underline{\sigma}^e$ is really not necessary. Rather we could calculate a new $\hat{\underline{u}}$ for each update of σ^e and combine the two iterations. Figure 6 shows the two different schemes. While not mathematically demonstrated, it was felt that such a scheme would further degrade convergence since

$\hat{\underline{u}}$ would undergo much larger variations. This issue should be considered in much more depth in continuing studies. This section will be concluded with a discussion of three topics, two very important, one included only for completeness. These topics are: convergence of the solutions, simplication due to ignoring the stress gradient terms of the constitutive equation, and equations used for independently interpolating the total deviatoric stresses in a mixed finite element method. We will discuss these topics in order.

Engelman et. al. [32] consider the problem of convergence of the general Navier-Stokes equation noting that Picard iteration converges more slowly than Newton-Raphson, but normally over a larger radius. They then treat the issue of accelerating convergence by employing guasi-Newton methods emphasizing Broyden-Fletcher-Goldfarb-Shanno updating. Such acceleration methods would enlarge the number of elements which can be economically treated in the solution scheme. Currently, however, this is not the problem with viscoelastic flow. As we will discuss in Section VI, the radius of convergence is the major issue, not the rate of convergence. Our study succeeded in obtaining solutions for $Ws \leq 0.01$ which could possibly be considered a trivial case. However, for the general flow geometries, we treated (in particular entry flow), the studies cited in the beginning of this section failed to achieve solutions even at that limit. Convergence therefore is the critical barrier to obtaining more general viscoelastic solutions. We did not pursue such extensions

in this study, but it is worthwhile to suggest a possible approach. Chung's [2] review of standard solution techniques is directly to this point. The radius of convergence can be widened by continuation methods. In particular, Chung suggests a multiple solution technique which combines incremental loading with Newton-Raphson corrections. Future effort in this field should investigate such an approach. We employed Picard iteration exclusively. Picard iteration should be tried as the top level, along with continuation methods. It is noted that both types of solution are amenable to the computer program used in this study.

We turn now to the simplications when the stress gradient terms are neglected. The terms themselves arise in the convection terms of the constitutive equation, i.e., $(\underline{u}\cdot\nabla)\underline{\sigma}$. For creeping flow similar terms were neglected in the Navier-Stokes equation and we know that for polymer melts, this is a good approximation. It is then obvious that we compare approximate magnitudes of $\nabla\underline{u}$ and $\nabla\underline{\sigma}$. For viscoelastic flows, we have already established that $\underline{\sigma}^e$ is on the order of $\underline{\sigma}^v$ and the gradients might be expected to be of equivalent nature. Therefore, we look at the comparison between the first derivative of $\underline{u}$ and the second derivative. It is known that even when $\underline{u}$ is discontinuous (as in the case of cross-channel flow of a screw extruder [9], the approximation at small distances from the singularities of $\nabla\underline{u}$ are quite good. This suggests that for creeping flow, a good approximation may be achieved when $(\underline{u}\cdot\nabla)\underline{\sigma}$ is neglected.

Equation IV.13 then becomes:

$$\underline{\sigma}^{e} = \lambda \underline{\underline{A}}(\underline{\sigma}^{v} + \underline{\sigma}^{e}) \tag{IV.24}$$

Evaluation of $\nabla\underline{\sigma}^{e}$ is eliminated and the Picard iteration becomes much more straightforward. An optional approach is to solve $\underline{\sigma}^{e}$ explicitly as:

$$(\underline{\underline{I}} - \lambda\underline{\underline{A}})\sigma^{e} = \lambda\underline{\underline{A}}\ \underline{\sigma}^{v} \tag{IV.25}$$

or

$$\underline{\sigma}^{e} = (\underline{\underline{I}} - \lambda\underline{\underline{A}})^{-1}\ \lambda\underline{\underline{A}}\ \underline{\sigma}^{v} \tag{IV.26}$$

Where $\underline{\underline{I}}$ is the unit matrix $\delta ij = \begin{cases} 1 & i=j \\ o & i\neq j \end{cases}$ $(i,j = 1,2,3)$

Equation IV.26 allows IV.5 to be written explicitly in terms of $\hat{\underline{u}}$ and the equation is a simple non-linear equation which can be solved with the numerical techniques discussed throughout the report. It is noted that although the explicit form makes the equations more straightforward, it is not expected that the radius of convergence (which is a function of λ) will be widened much. However, at the early stages of research efforts, particularly in applying continuation methods, this equation seems to offer promise.

Finally, the mixed method of solution is briefly discussed for sake of completeness. Following Kawahara's approach [28], we set up the simultaneous equations for steady state in indicial notation:

$$\rho\, u_j\, u_{i,j} + P_{,i} - \sigma_{ij,j} = o \tag{IV.27a}$$

$$\sigma_{ij} + \lambda(u_k \sigma_{ij,k} - u_{i,k}\sigma_{kj} - u_{j,k}\sigma_{ik}) - \mu(u_{i,j} + u_{j,i}) = o \tag{IV.27b}$$

Both IV.27a and IV.27b are non-linear; we write the finite element equations:

$$\left\{ \int_v (\underline{\underline{N}}^T \rho (\nabla \cdot (\underline{\underline{N}}\ \hat{\underline{u}})^T)^T \underline{\underline{N}} + (\underline{m}^T\ \underline{\underline{B}})^T \alpha \underline{m}^T\ \underline{\underline{B}})\, dv \right\} \hat{\underline{u}} + \left\{ \int_v \underline{\underline{N}}^T \underline{\underline{B}}^{*T} dv \right\} \hat{\underline{\sigma}} = \underline{f} \tag{IV.28a}$$

$$\left\{ \int_v (\underline{\underline{N}}^{*T} \lambda \nabla (\underline{\underline{N}}^* \hat{\underline{\sigma}}) \underline{\underline{N}} - \underline{\underline{N}}^{*T}\ \underline{\underline{D}}\ \underline{\underline{B}} - \underline{\underline{N}}^{*T} \underline{\underline{Q}}\ \underline{\underline{N}})\, dv \right\} \hat{\underline{u}} + \left\{ \int_v \underline{\underline{N}}^{*T} \underline{\underline{N}}^* dv \right\} \hat{\underline{\sigma}} = o \tag{IV.28b}$$

Where the asterisk indicates the trial function for the stress interpolation.

The solution to IV.28 can be seen clearly if we form a typical equation in matrix form:

$$\left[\begin{array}{c|c} \begin{array}{l} \underline{\underline{N}}_i^T \rho (\nabla \cdot (\underline{\underline{N}}\ \hat{\underline{u}})^T)^T \underline{\underline{N}}_j \\ + (\underline{m}^T \underline{\underline{B}}_i)^T \alpha \underline{m}^T \underline{\underline{B}}_j \end{array} & N_i^T B_j^{*T} \\ \hline \begin{array}{l} \underline{\underline{N}}_i^{*T} \lambda \nabla (\underline{\underline{N}}^* \hat{\underline{\sigma}}) \underline{\underline{N}}_j \\ -N_i^{*T}\ \underline{\underline{D}}\ \underline{\underline{B}}_j \\ -N_i^{*T} \underline{\underline{C}}\ \underline{\underline{N}}_j \end{array} & N_i^{*T} N_j^* \end{array} \right] \left[\begin{array}{c} \hat{u}^1 \\ \hat{v}^1 \\ \\ \hline \hat{\sigma}_{xx}^1 \\ \hat{\sigma}_{yy}^1 \\ \hat{\sigma}_{xy}^1 \end{array} \right] = \left[\begin{array}{c} F_1^1 \\ F_2^1 \\ \\ \hline 0 \\ 0 \\ 0 \end{array} \right] \tag{IV.29}$$

(In equations IV.29, the integrals are implied.)

In IV.29, the superscript in the column vectors indicate the *node number so that this relation is repeated for each of* the nine nodes. i and j indicate the row and column in the assembled array (for IV.29 i=j=1). The array is partitioned accordingly so that the upper left corner is 2 x 2, upper right corner is 2 x 3, lower left corner is 3 x 2, lower right corner is 3 x 3. All matrices in IV.29 have been previously defined with the exception of $\underline{\underline{Q}}$ which is:

$$\underline{\underline{Q}}=\lambda\begin{bmatrix} 2(\underline{\underline{N}}^{*}\hat{\sigma}_{xx})\frac{\partial}{\partial x}+2(\underline{\underline{N}}^{*}\hat{\sigma}_{xy})\frac{\partial}{\partial y} & 0 \\ 0 & 2(\underline{\underline{N}}^{*}\hat{\sigma}_{yy})\frac{\partial}{\partial y}+2(\underline{\underline{N}}^{*}\hat{\sigma}_{xy})\frac{\partial}{\partial x} \\ \underline{\underline{N}}^{*}\hat{\sigma}_{yy}\frac{\partial}{\partial y}+\underline{\underline{N}}^{*}\hat{\sigma}_{xy}\frac{\partial}{\partial x} & \underline{\underline{N}}^{*}\hat{\sigma}_{xx}\frac{\partial}{\partial x}+\underline{\underline{N}}^{*}\hat{\sigma}_{xy}\frac{\partial}{\partial y} \end{bmatrix}$$

These equations when fully assembled yield a set of linear equations of order 5_p, where $_p$ is the number of nodes. For a nine node element then the order of equations is 45. The number of variables for the whole domain then would be 45n-m with n being the number of elements and m the number of shared nodes. It can be seen that it does not take many elements to generate a very large computer region to solve the equations. While the above analysis was conducted and subroutine ELMTØ6 written for the problem solution, no flow cases were run in this study. Future work may implement subroutine ELMTØ6.

V. COMPUTER IMPLEMENTATION

In this section we will discuss the major aspects of the finite element program, the calculational procedures, and the input/output.

The source program was a modified version of the Finite Element Analysis Program (FEAP) written by Prof. R. L. Taylor at the University of California, Berkeley, and published in Chapter 24 of [1]. The modifications have been made by Prof. David Roylance of the Massachusetts Institute of Technology to accommodate polymer melt flow [9]. These modifications are largely: (i) addition of a power law flow rule, (ii) addition of a temperature dependent viscosity, (iii) alteration of matrix algebra operations, and (iv) addition of an axisymmetric capability. The rationale for using this model is given in [9]. The current effort included reviewing the source program to insure correctness, and modifying it to include a viscoelastic flow option. Currently the program is two-dimensional (rectilinear or axisymmetric) and steady state.

The program establishes a dynamic storage vector at the outset which is partitioned to store all input data (node coordinates, element node numbers, etc.), global data (stiffnesses, loads, etc.) and output data (velocities). Other features are a linear interpolation mesh generation scheme, an active equation solver and a macro command language which controls the solution execution. The macro commands and their meaning

are listed in Table 24.12 of [1].

Upon construction of the architecture of the problem, calculations required for a specific command (such as forming the tangent stiffness matrix) are made in a library of element subroutines. Subroutine PFORM steps through the n elements by forming element arrays from global data and passing the arrays to the element routine. Subroutine ELMTØ5 is a general 2D penalty method solution of the Navier-Stokes equation written by Frecaut [16]. This is the element subroutine modified for the viscoelastic flow.

The basic source program flow chart is given in Figure 7. To modify this program for viscoelastic flow, three basic changes were made. First was to flag the problem as visco-elastic and read material data. This was done in subroutine DFMTRX. The card reading format after input macro command MATE was changed to the following:

CARD 1 Format (I5, 4X, I1, 17A4)

CARD 2 Format (4I5, F1Ø.Ø)

CARD 3 Format (I5, 7D1Ø.4)

Card one reads the material set number in columns 1-5 (in all cases only one material set is used and therefore this is 1), the element type in column 10 (5 for ELMTØ5) and the problem description in the remaining columns. Card two reads the flow type in columns 1-5 (1 = plane flow, 3 = axisymmetric flow), a flag (N1) for thermal coupling in columns 6-10 (Ø = isothermal, 1 = thermally coupled), a flag (K2) for viscoelasticity in

columns 11-15 (1 = simple viscous, 2 = power law viscous, 3 = White-Metzner Viscoelastic, 4 = DeWitt Viscoelastic, 5 = Rivlin Ericksen Viscoelastic), a flag (N3) for the time domain in columns 16-20 (1 = steady state, 2 = unsteady), and the power law coefficient (P4) in columns 21-30. P4 must be included and for simple viscous material P4 = 1.Ø (which was the case treated exclusively in this study).

Card three reads the Gauss integration order (L) in columns 1-5 (2 = 2x2), the penalty coefficient (XLAM) in columns 6-15, the viscosity coefficient (XMU) in columns 16-25, the density (RHO) in columns 26-35, the viscoelastic shear modulus (G) in columns 36-45, the thermal conductivity (XK) in columns 46-55, the specific heat (C) in columns 56-65, and the work-to-heat conversion factor (HEAT) in columns 66-75. The program is written so that when data is not required for the specific problem (e.g. linear, steady, isothermal, inelastic flow) those columns may be left blank. In card three then only columns 1-25 need be included.

The second change was to add algorithms in ELMTØ5 for the calculation of the elastic stresses according to equations IV.13. The last change presented the major difficulty: the calculation of the elastic stress gradients according to equations IV.19. As noted in the previous section, no scheme existed for making calculations with variables from different elements. In order to solve IV.19, however, this was necessary. The approach taken was to define common arrays YY(I,J,N), ESIG1(I,J,N)

ESIG2(I,J,N),ESIG3(I,J,N),ELAS1(I,J,N),ELAS2(I,J,N),ELAS3(I,J,N), and BOSIG(I,J,N). YY is the global coordinate (J=1,2) of the Gauss points (I=1,4). ESIG1, ESIG2, and ESIG3 are σ_{xx}, σ_{yy} and σ_{xy} respectively at the Gauss points (I=1,4) at iteration J=K, K+1. ELAS1, ELAS2, and ELAS3 are the gradients (J=1,2) of $\underline{\sigma}^e$ at the Gauss points (I=1,4). BOSIG is the elastic stress (J=1,3) at the boundary, at the corner node (I=1,4). In all the arrays N is the element number. In PFORM, N is passed as common through ELMLIB and ELMTØ5 and it is therefore possible to conduct the calculations between the two subroutines PFORM and ELMTØ5. The gradients of the three stress components at the Gauss points are first solved for all the elements assuming they are a boundary element on all sides. A searching scheme is then affected which compares the nodes of all the other elements. When two elements are found in the correct location, the elastic stress gradients are replaced at that Gauss point. If adjacent elements are not found, the element is on a boundary and that Gauss point is left unchanged. During the Richardson/Jacobi iteration, the elastic stress gradients then are calculated in PFORM and these values used in ELMTØ5 to calculate the updated values of the elastic stress at the K+1 iteration. This iteration is conducted 20 times unless convergence is achieved beforehand. The program then continues in a normal manner.

The listings of the major subroutines written to accomplish the modifications are included in Appendix 3. The subroutines are in order listed ELMTØ5, ELMTØ6, ESHAP, PFORM, CMATRX, and

FPSIG. ELMTØ6 is the subroutine written for interpolate total deviatoric stresses in a mixed method. ESHAP is the calculation of the second derivatives and FPSIG is a new routine written to print viscous and elastic stresses at the Gauss points. CMATRX is the subroutine which forms the Q matrix in ELMTØ6.

VI. CALCULATION RESULTS

Four flow geometries were treated as shown in Figure 8 (along with the boundary conditions): Cross Channel Flow, Plane Couette Flow, Entry Flow, and Step Flow. Table 1 shows the computer run matrix. The input data sets for runs 1, 3, 4, 6, 13, and 20 are included as Appendix 4. Results are discussed below for each of the four problems treated. For all cases, the viscosity coefficient was taken to be 790 poise. This was the value selected by Roylance [9] in previous studies. His reasons were unrelated to the work in this study, but we chose to use the same value for comparison purposes. With more reasonable values (10^4), we would only expect to see higher stresses, but no change in the velocity fields.

CROSS CHANNEL FLOW

The solution of creeping flow, circulating in the transverse plane of channel, for a viscous fluid is well known (e.g. [9]). At steady state, the circulation is uniform with a vortex center at mid-height, towards the vertical boundary on the right in Figure 8a. This study looked at the consistency of reproducing this flow with 9 node and 8 node elements and the effects of a finite fluid elasticity. Secondary eddies and screw power requirement changes were considered to be demonstrable effects of elasticity.

Figure 9 shows the velocity vector flow field for run 1 (linear case). Results are identical to [7], different from [9]. This is due exclusively to the specified boundary condition at the upper corners of the channel. For our boundary conditions, the vortex center is at the mid-width of the channel near the 2/3 height section.

The velocities calculated for the nodes of elements 7, 9, and 15 by the 18 element 9 node and 18 element 8 node case are compared in Figure 10. Note that a significant difference occurs in the direction of the resultant velocities in element 7 and the magnitude in element 15 (a 20% lower horizontal velocity is predicted in the middle nodes of element 15 by the 8 node model). When the results of the 72 element, 8 node case are examined (run 3) the 9 node model is found to be uniformly closer. The velocity field is, therefore, predicted much better by the 9 node elements for the same number of elements.

Let us now make a practical application. The power per unit area required of a single flight screw extruder to create this circulation is the shear stress in the fluid times U_B (the relative barral velocity). If we approximate this as the average element shear stress σ_{xy} times the average velocity in the element, we have the following for element 15:

	9 NODE 18 ELEM	8 NODE 18 ELEM
$\overline{\sigma}_{xy}$ (dynes/cm^2)	0.22×10^6	0.2×10^6
$\overline{u}$ (cm/sec)	-50	-52.5
$\dot{w}$ (dyne-cm/cm^2-sec)	1.08×10^7	1.05×10^7

We can conclude that the 9 node elements yield more accurate node velocities, but when average properties are sought, such as the power or torque required for the screw design, both models give approximately the same results for equivalent meshes. This, of course, is expected since the finite element equations satisfy equilibrium over the entire region. However, on a local scale (which we are also interested in) the above justifies our earlier preference for the 9 node elements.

From Hughes data [7], the effects of increasing the Reynold's number (Re) is to shift the vortex center toward the right-hand boundary. This was investigated for one case by choosing the density of polyphenelenesulfide (1.6 gm/cm^3). Combining this with the other characteristic numbers of the cross-channel flow problem, we obtain $Re = \frac{\rho UL}{\mu} = 0.41$. Including the convection non-linearity for this Re we found no discernible perturbation to the velocities or stresses, thus confirming the validity of the "creeping flow" analysis.

For the single viscoelastic case for which the solution converged (Ws = 0.02) the velocity field again did not vary appreciably. Figure 9 can, therefore, be considered correct for this level of elasticity. To look at the stress effects,

we make the same calculation for the specific power as above yielding:

	NEWTONIAN	VISCOELASTIC (Ws=0.02)
$\bar{\sigma}_{xy}$ (dynes/cm^2)	0.22×10^6	0.22×10^6
$\bar{u}$ (cm/sec)	-50	-50
$\dot{w}$ (dyne-cm/cm^2-sec)	1.08×10^7	1.08×10^7

Within roundoff error, the two flows are identical (maximum σ_{xy} deviation was 1%). A second comparison is available in Figure 11 where the pressure is plotted at the mid-height as a function of the cross-channel (transverse) station. Again the viscoelastic flow is coincident with the Newtonian case. Within the range of calculations achieved in this study therefore ($Ws \leq 0.02$), there are no effects of viscoelasticity manifested. We do observe, however, that the stresses calculated (~1% variation) are consistent with the Ws suggesting accuracy of the computer model when convergence is achieved.

PLANE COUETTE FLOW

Plane Couette flow was selected for the fundamental evaluation of the computer model. This is through the relation presented by Middleman [26]:

$$S_R = \lambda\dot{\gamma} \qquad \text{(VI.1)}$$

where S_R is the recoverable (elastic) shear stress:

$$S_R = \frac{\sigma_{xx} - \sigma_{yy}}{2\sigma_{xy}}, \qquad \text{(VI.2)}$$

λ is the relaxation modulus and $\dot{\gamma}$ is the steady, simple shear flow strain rate. The flow is enforced by specifying a linear variation of the horizontal velocity between two plates, one stationary, the other moving at a constant velocity as shown in Figure 7b.

Run 7 was the Newtonian case to validate the problem. In this case, σ_{xx} and σ_{yy} should be identically zero and σ_{xy} constant throughout the field domain. This was the result of the calculation.

For the viscoelastic case (Run 10), all the normal stresses are elastic while from equations IV.13, with $v = \frac{\partial}{\partial x} = o$ only $\sigma_{xx}{}^{e}$ is finite. Therefore, we should observe the following:

$$S_R = \frac{\sigma_{xx}{}^{e}}{2\sigma_{xy}{}^{v}} = \lambda\dot{\gamma} \equiv \text{constant} \qquad \text{(VI.3)}$$

For a unit height between sliding plates we have $\dot{\gamma} = U_B$ so that:

$$\sigma_{xx}{}^{e} = 2\lambda U_B\ \sigma_{xy}{}^{v} \qquad \text{(VI.4)}$$

The computer results are for $\lambda = 0.0002$, $U_B = 100$ cm/sec (Ws = 0.02):

$$\sigma_{xx}{}^{e} = 3.16 \text{ KPa}, \quad 2\lambda U_B \sigma_{xy}{}^{v} = 3.16 \text{ KPa}.$$

The equation is identically satisfied. This, of course, is encouraging for future work to increase the radius of convergence for higher Ws numbers.

ENTRY FLOW

The entry flow problem for viscoelastic fluids has not been successfully calculated by finite element methods in the past, due to severe numerical convergence problems. As a first step, Run 11 was accomplished for linear flow according to the boundary conditions specified in Figure 7c. A discussion of these boundary conditions is in order.

Rather than a constant horizontal velocity at the inlet to the reservoir (upstream channel), a more accurate analysis would specify fully developed flow. Middleman [26] presents this for flow between parallel plates (for a Newtonian fluid as):

$$u = \frac{B^2 \frac{\partial P}{\partial x}}{8\mu L}\left[1 - \left(\frac{2y}{B}\right)^2\right] \qquad \text{(VI.5)}$$

where B is the channel height

and L is the channel length

(all other variables retain their earlier definition).

For a White-Metzner fluid, the plane-Poiseville flow would be solved by adding the elastic stresses to the momentum equations. Perera [27] did this for a 4 constant Oldroyd fluid and solved the resulting second-order differential equation for u(y) by Newton-Cotes integration. With equations of the type specified in VI.5, we can solve the pressure loss $\frac{\partial P}{\partial x}$ due to inlet and outlet. In addition White [33] cites the additional pressure losses due to entrance and exit of the dies. It is these boundary conditions that would be more realistic in treating the entry flow problem (velocity according to VI.5

at one end, ΔP at the other). With the formulation specified in this work, it was expected that the flow field would behave quite differently from the classical converging type. Since we did not have data on die pressure losses, however, the initial calculations were made on the basis of the boundary conditions given.

When fully developed conditions are specified, both upstream and downstream of the entrance region the flow is known to be stable up to relatively high Ws numbers. At Ws around one secondary vortex patterns arise which are generally ascribed to increasing elastic stresses generated in the shearing/elongational flow (White [33] implies that elongational flow is important and we, therefore, conclude that the Rivlin-Ericksen fluid simplified for viscometric flow is a questionable model). This flow behavior is documented in Figure 12 which shows experimental behavior noted by White [33] as a function of Ws and calculations of Perera [27] for Ws = 0.6.

The calculated velocity field for the boundary condition specified in Figure 8c is shown in Figure 13. Although the mesh is very coarse, it appears that the flow is unstable for these conditions. The viscoelastic calculation (Run 12, Ws = 0.01) exhibited identical behavior. Because of this poorly behaved flow field, the calculation was repeated using the fully developed flow boundary conditions. The results are shown in Figure 14. The specific boundary conditions were established in the following manner. The excess pressure losses described by White [33] were ignored (this will affect

the calculation however). At y = 0 (y measured from the mid-height of the channel) equation IV.5 is

$$u = \frac{B^2 \Delta P}{8\mu L} \tag{IV.6}$$

For the two channels, there would be a total pressure loss of $\Delta P_T = \Delta P_I + \Delta P_O$ if the flow was fully developed. Therefore

$$\Delta P_T = 8\mu \quad \frac{L_I u_I}{B_I^{\,2}} \quad + \frac{L_O u_O}{B_O^{\,2}} \tag{IV.7}$$

For our geometry $L_I = L_O = B_I = 1$, $B_O = \frac{1}{2}$, and μ = 790 poise. There are three unknowns in equation IV.7. However, rather than specifying two of the three, we merely let $\Delta P_I = \Delta P_O$ (which has the same effect) and specified u_O. This permits the calculation of u_I and thereby calculation of ΔP_T. This ΔP_T was established as the inlet traction P_I and the outlet was atmospheric $P_O = 0$. This yields the value of $P_I = 6.4\mu$. The pressure is converted to the virtual "work" equivalent node forces by the relationship

$$F_x^C = \frac{1}{3} H \; P_I$$

$$F_x^M = \frac{4}{3} H \; P_I \tag{IV.8}$$

where the superscript denotes the element node (c - corner node, M = mid-side node) and H is half the element height at the nodes. (See Frecault [16] for the details of virtual "work" equivalence calculations; equation IV.8 are valid for 8 node and 9 node plane quadrilateral elements). (In the actual boundary node forces, the corner nodes are loaded with

$F_x^c = \frac{2}{3} H P_I$ for uniform meshes since the node is shared by adjacent elements. Only the vertical velocities at the boundaries (v=o) now must be specified. The mid-height horizontal velocity will not be the value used in the calculation, but the flow will be fully developed.

Comparing Figures 13 and 14, we see that although the behavior is somewhat improved by the fully developed flow case, there is still major error in the flow field and even flow reversal. This is felt to be attributable entirely to the coarseness of the mesh, particularly near the entry corner. A finer mesh case was not constructed to test this hypothesis. It is recommended that future work include this refinement.

Notice that symmetry was not employed to reduce the number of elements. This was due to the difficulty of specifying boundary conditions on the plane of symmetry. The first conditon is v=o, but the other boundary condition is not so straightforward. We know that $\frac{\partial u}{\partial y}$ and $\frac{\partial v}{\partial x}$ are zero at the mid-height, but in general $\frac{\partial v}{\partial y}$ is not zero within the reduction region. This, of course, is a statement that the one dimensional lubrication theory is not valid. Since the pressure now changes across the channel height the pressure in the x direction can no longer be specified as a linear function of x. Therefore, the nodal loading in the x direction is unprescribed as well as the velocity u. Of course, this could be resolved by adding the condition of no mass flow across the plane of symmetry. We chose not to accomplish this at the penalty of doubling the number of elements.

If the flow were one dimensional, the pressure would vary linearly with the length and the velocity would be constant in each of the two sections (plane Poiseulle flow). Figure 15 shows the deviation from this case.

It was noted in examining the stresses in Runs 11 and 12 that the difference was much larger than expected for the low Ws. However, a thorough evaluation was not conducted because it was felt that the differences were an artifact of the calculations due to the following: (i) the velocity fields were erratic as previously mentioned due to the coarse grid, (ii) the boundary conditions of constant inlet velocity gave rise to poorly behaved pressure variations even for the Newtonian case, and (iii) the solution convergence for the non-linear problem was still poor at 30 iterations. It is noted in passing, however, that as the solution procedure is improved, it is exactly these types of variations which are being sought.

STEP FLOW

This geometry was selected as the beginning step toward an analysis of flow past an obstruction such as would be the case if pins were added to the cavity to form holes in the molded part. With the boundary conditions specified in Figure 8d, the results were very similar to those discussed for entry flow. A discussion of the computer calculations will therefore not be included in this report. It is noted, however, that there is still negligible differences between Runs 15 (linear Newtonian) and 16 (convection Newtonian).

This begins to address the issue of "Stokes paradox" and the necessity of including convection, even for low Re, for obstructed flow. The paradox is that in two-dimensional flow no analytic solution exists for the linear equation which matches the boundary conditions at the surface of the obstruction and at large distances away from it [34]. Batchelor [35] shows that when the distance from the obstruction (or a boundary) is on the order of ℓ/Re (where ℓ is a characteristic dimension of the obstruction) the convection stresses (inertia) may become of equal importance to the viscous stresses. Analytically this correction is known as O'seen's improvement. Again as the model described in this report is refined, the adequacy of the "creeping" flow analysis must be examined in light of this issue.

VII. MODEL EVALUATION

It is worthwhile to complete a qualitative evaluation of the computer model before this report is concluded. Figure 16 presents a diagram of a complete model for a real injection molding process. The Figure emphasizes those elements included in this study. Since we achieved numerical convergence for Ws $\leq$ 0.01 it must be concluded that a non-Newtonian power law fluid analysis would be as good an approximation as the viscoelastic model used herein. If future work does not improve this convergence region (at least to Ws $\geq$ 0.5) the numerical analysis would seem to be as good without including elastic effects. Also finite difference methods have succeeded in obtaining solutions up to Ws = 0.6 [27] and it may therefore be advisable to develop these techniques for application to the gyroscope manufacturing.

The model is steady state and includes no free surfaces such as would occur during the mold filling period. Therefore, it can only be used in regions such as the extruder, nozzle, sprue, runner and gate. Unless unsteady, free surface terms are included, this model is not applicable to the mold filling itself. But the power required to supply a nozzle with a given rate of flow is certainly within the capability of the model. Also the state of the bulk material as it passes through the gate can be determined by use of this model. Any damage due to high stresses or thermal degradation in these regions can be analyzed with the model. It is noted that although there

will be a finite elastic stress which the polymer can sustain before flowing completely plastic (viscous plus the elastic limit stress), there is no yield stress built into this model. Therefore, while the model will predict continually increasing stresses, judgement must be exercised as to the real elastic capacity of the fluid.

The current status of the coupled heat transfer capability of the model is the adiabatic model developed by Roylance [9]. Extension to a complete non-isothermal boundary analysis can be implemented without too much difficulty.

We have noted that major modifications are necessary to evaluate the mold filling itself (only pressure and filling rate can currently be analyzed). Also within the mold, the cooling stage of the molding process can not be analyzed because of the absence of a solid thermomechanical viscoelastic model.

However, if an initial state can be established for the cooling process such a model could be developed.

The mold filling process itself can take the approach of a constant flow rate at the gate once free surface effects are added to the model. This is the approach used in [11]. The free surface analysis is most clearly discussed in [4] where the front displacement is calculated over some interval of time assuming a constant velocity of the boundary elements node points. The surface traction on the flow front is zero normal to the surface and the material surface tension tangential to the surface.

From Section V, we can discuss the approach to improving the viscoelastic case. To assess the maximum radius of convergence of the momentum equation, it is adequate to neglect $\underline{u} \cdot \nabla \underline{\sigma}$ and use equation IV.26. Since Newton-Raphson iteration generally converges for the Navier-Stokes equations well above Re = 25, it should be verified that convergence is achieved with the current numerical approach for Ws = 25. With this step accomplished, $\underline{u} \cdot \nabla \underline{\sigma}$ can be added and the continuation method used. The effective technique should employ incremental loading with Newton-Raphson corrections. Let us discuss this a little further. Since we are using direct (Picard) iteration on the elastic stress terms, let us rewrite equation IV.11 as:

$$\underline{\underline{K}}\ \hat{\underline{u}} = \underline{f} - \underline{K}^{e} \qquad \text{(VII.1)}$$

Since Picard iteration is a single point scheme, (i.e., the initial value of $\underline{\underline{K}}\ \hat{\underline{u}} + \underline{K}^{e} - \underline{f}$ is always used rather than updating in the Newton-Raphson scheme, see Figure 17) we can attempt to increment this point. Therefore, instead of solving VII.1 directly, we solve:

$$\underline{\underline{K}}\ \hat{\underline{u}} = \theta(\underline{f} - \underline{K}^{e}) \qquad \text{(VII.2)}$$

where $0 \leq \theta \leq 1$. With the solution to VII.2 converging for sufficiently small numbers of θ we can update the initial selection of $\hat{\underline{u}}$ by incrementing θ. For example, let $\theta = 0.1$ then in the first increment the first value of $\hat{\underline{u}}$ is :

$$\hat{\underline{u}}^{o} = \underline{\underline{K}}^{-1}\ \theta \underline{f}$$

We then iterate with $\underline{\underline{K}}\ \underline{u}^{s+1} = \theta[\underline{F} - \underline{K}^e(u^s)]$.

When convergence is achieved then we increment θ to $2\theta = 0.2$.

Then

$$\hat{\underline{u}}^o = \underline{\underline{K}}^{-1}[2\theta\underline{F} - \underline{K}^e(u^{s+1})]$$

Therefore, the initial guess is improved by the correction $\underline{K}^e(u^{s+1})$. It is noted that this technique is different from the normal continuation methods where the non-linear equation is always of the form: $K(u)u=f$. While no mathematical analysis has been conducted on this proposed technique, it appears to offer promise.

This deviation in the classical incremental load method is only necessary when the stress gradient terms are included in the viscoelastic constitutive model. Therefore, when the model undergoes its first revision with $\underline{u}\cdot\nabla\underline{\sigma}$ neglected we write $\underline{\sigma}^e$ explicitly and if convergence fails the classical incremental load methods described in [1] and [2] should be employed.

VIII. CONCLUSIONS

This report has dealt primarily with the additional mathematics required to incorporate elasticity in the deviatoric stresses developed in flowing polymer melts. Implementation of the equations within an existing Finite Element Computer routine was then shown. From these analyses we can make the following conclusions:

- The direct Picard Iteration Converges within a radius of $Ws \leq 0.01$.
- For cross channel flow and entry flow "creeping" solutions are very accurate for typical polymer extrusion Reynold's numbers ($Re < 0.4$).
- For the Weissenberg numbers which yielded convergence, no appreciable effects on the flow were noted.
- The programming technique of passing data between elements by common memory appeared to be effective.
- When convergence was achieved. the calculated values of elastic stresses were consistent and reasonable.
- The penalty method of incompressible flow appears to yield good results for viscoelastic fluids.
- The radius of convergence was consistent with previous finite element calculations.
- The radius of convergence can certainly be improved by finite difference calculations as evidenced by Perera [27].
- Without improvement, the only computer options which should be used in evaluating polymer fluids are Newtonian and power law viscous (isothermal and adiabatic).

- Techniques of improving the viscoelastic model have been proposed which offer great potential.
- For 24 element problems, the computer cost for runs requiring 30 iterations was $100.00.

IX. RECOMMENDATIONS

It is felt that the work performed in this study offers potential for useful follow-on effort. In particular, there are three areas of development. First, the analysis of the complete flow problem is vital. While the gyroscope fabrication is new, the need for numerical evaluation in the molding process is not. The work at Cornell [11] demonstrates this fact. In that effort, the various regions of flow are being tied together. A similar approach is required for the finite element modeling. A model which connects the flow within and out of the extruder, through the various conduits, and into the mold cavity is an important development which should be pursued.

Direct extensions of the work addressed in this study are also important. The approach should be: (i) ignore stress gradients in the constitutive equation and conduct direct calculations, (ii) add stress gradients along with continuation solution methods of non-linear equations. Even if future work with constitutive models which include stress gradients are unsuccessful, it is felt that the equation with some elastic stresses will be a big improvement over Newtonian or power law fluids.

Finally as efforts one and two above progress, there is a need to conduct rheology experiments which will determine properties of the fiber-filled polymers being used in the gyroscope fabrication. These data are required to correlate with the velocities and stresses predicted by finite element equations.

The three categories are listed below:

- Model complete flow history from extruder to mold cavity.
- Refine viscoelastic model.
- Conduct rheology experiments of appropriate polymeric materials.

MOLDED INERTIAL GYRO

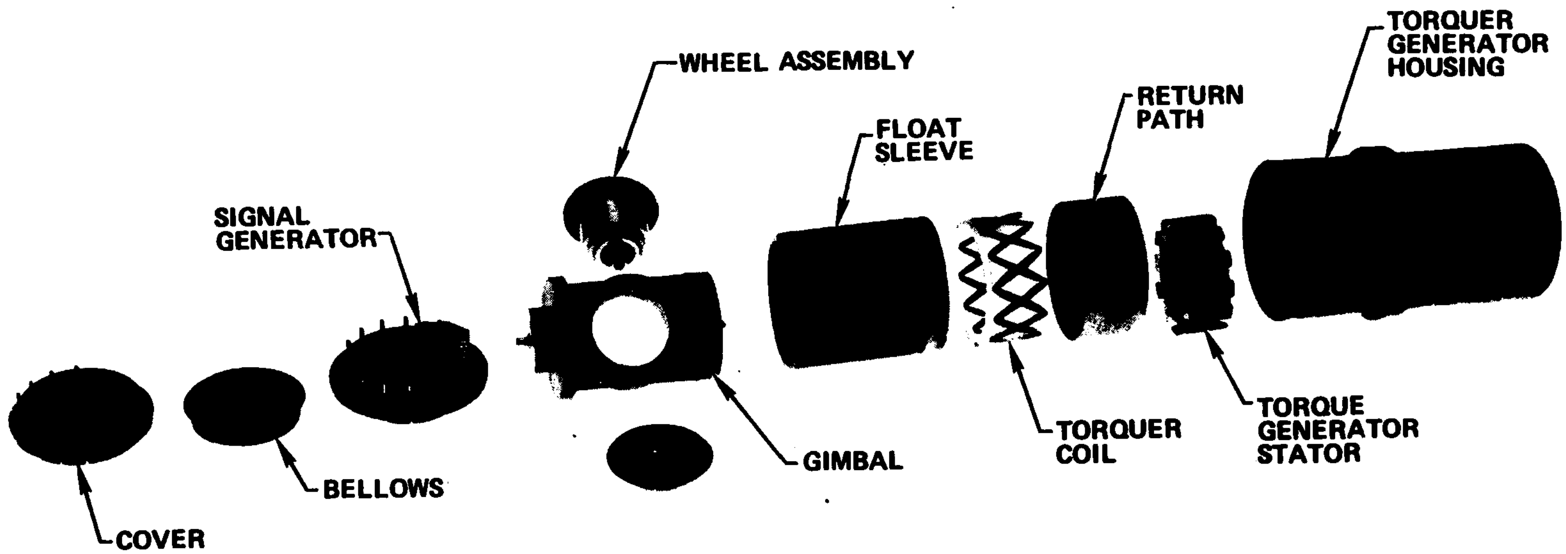


Figure 1 Piece Part Assembly of a Plastic Gyroscope

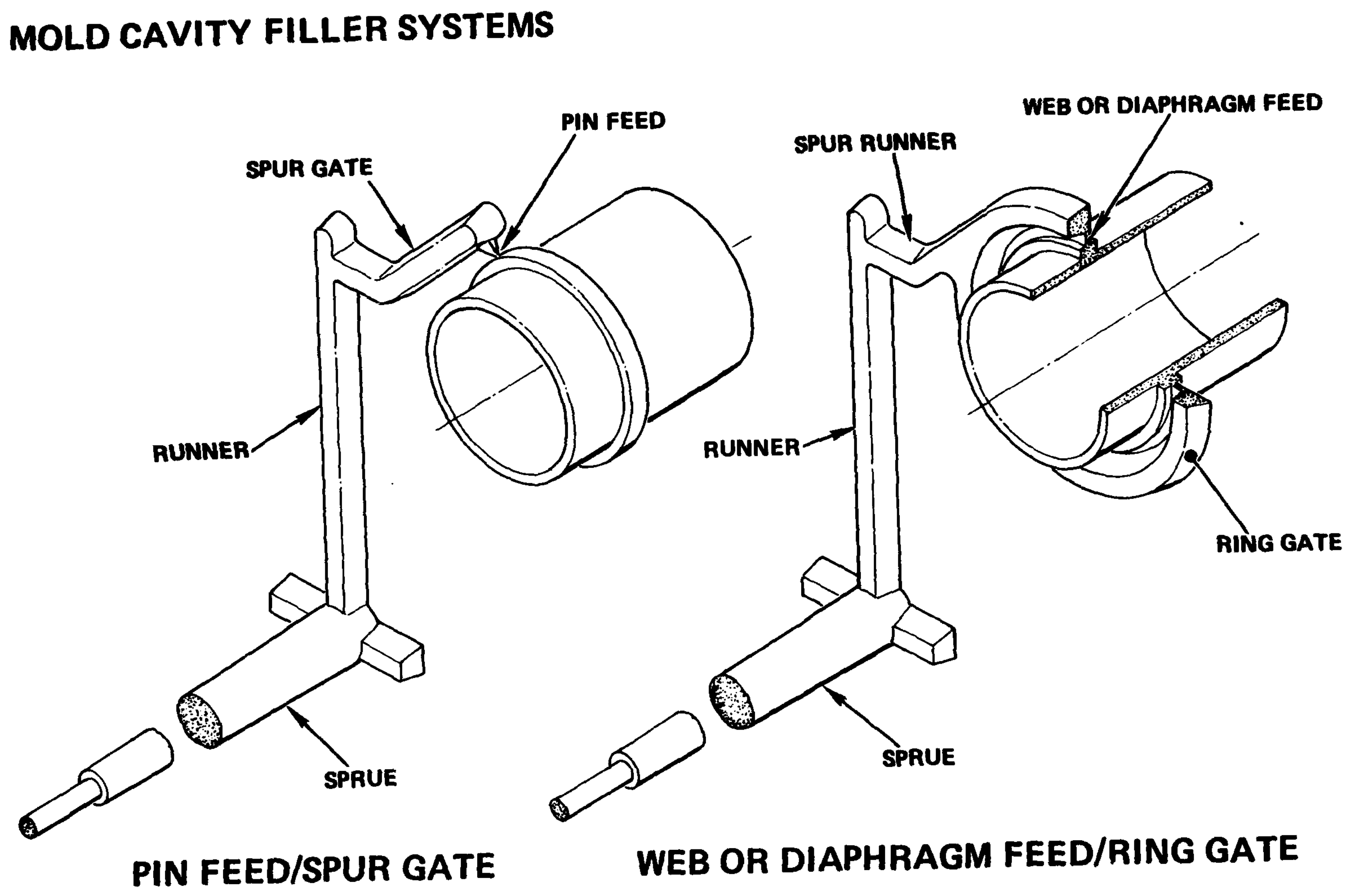


Figure 2 Typical Injection Molding Process

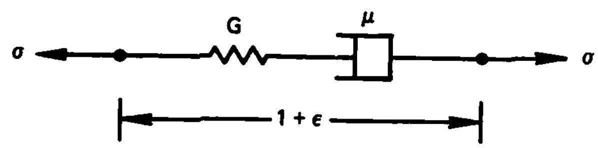


Figure 3 Fluid Maxwell Element

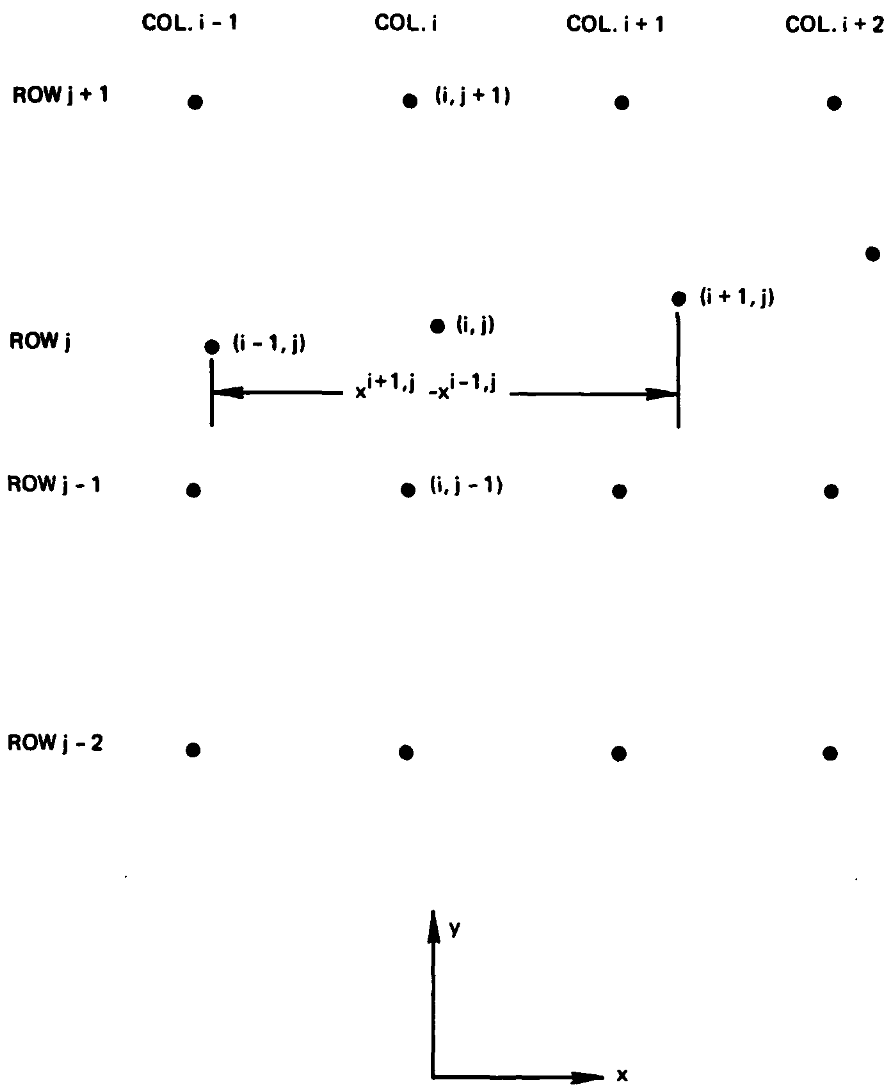
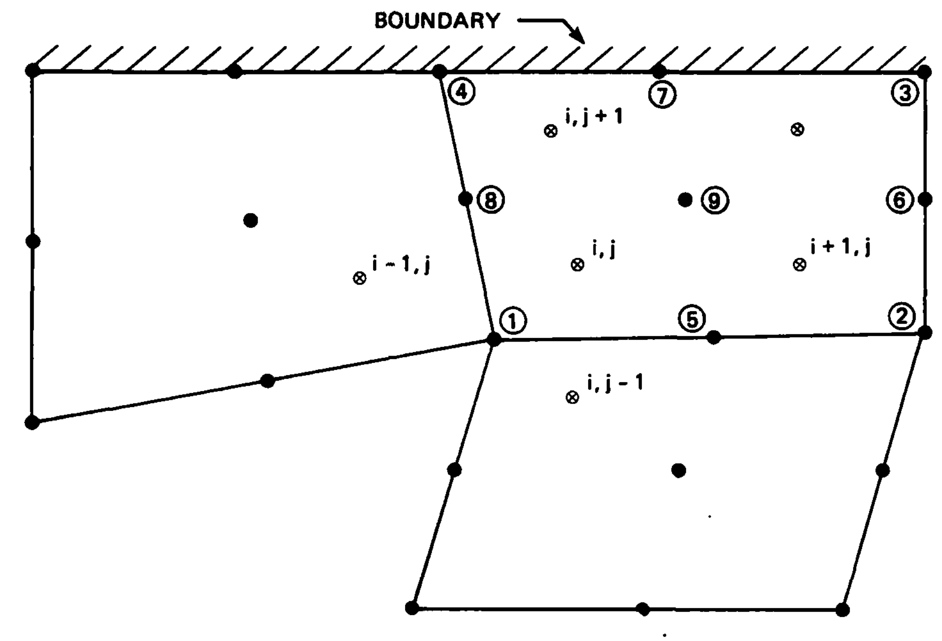


Figure 4 Non-Uniform Molecule Mesh for Solving Stress Gradients



○ NODES
⊗ GAUSS POINTS

EQUATION IV-18 USED AT $\oplus^{i,j}$
EQUATION IV-23 USED TO CALCULATE $\underline{\sigma}_e^{N.P}$ AT ④
EQUATION IV-19 MODIFIED AT $\oplus^{i,j+1}$ AS FOLLOWS:

$$x^{i,j+1} = x^{④},\ y^{i,j+1} = y^{④},\ \sigma^{i,j+1} = \sigma^{④}$$

(Note: Superscripts are indexed at each Gauss point so that $x^{④}$ is $x^{i,j+2}$ referred to Gauss point 1 whereas it is $x^{i,j+1}$ referred to Gauss point 4)

Figure 5 Calculation of Elastic Stresses at Gauss Points

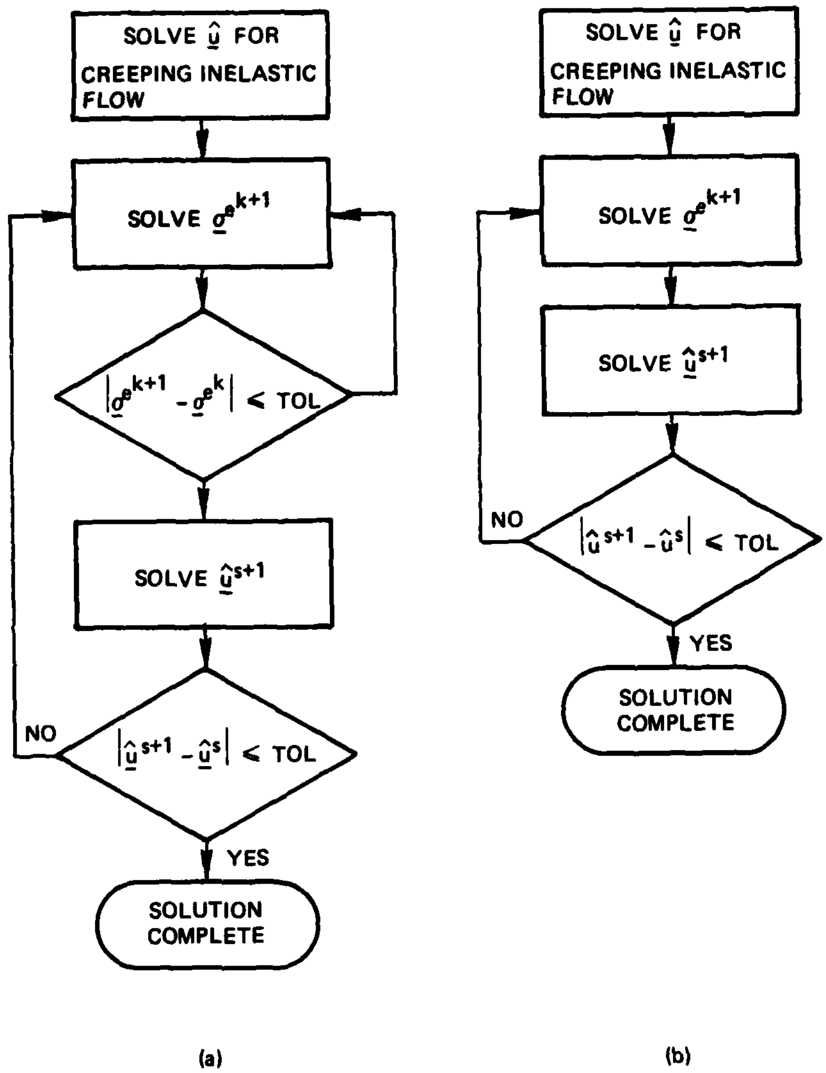


Figure 6 Iteration Schemes for the Solution of Creeping, Viscoelastic Finite Element Equations: (a) Nested Iteration (b) Combined Iteration

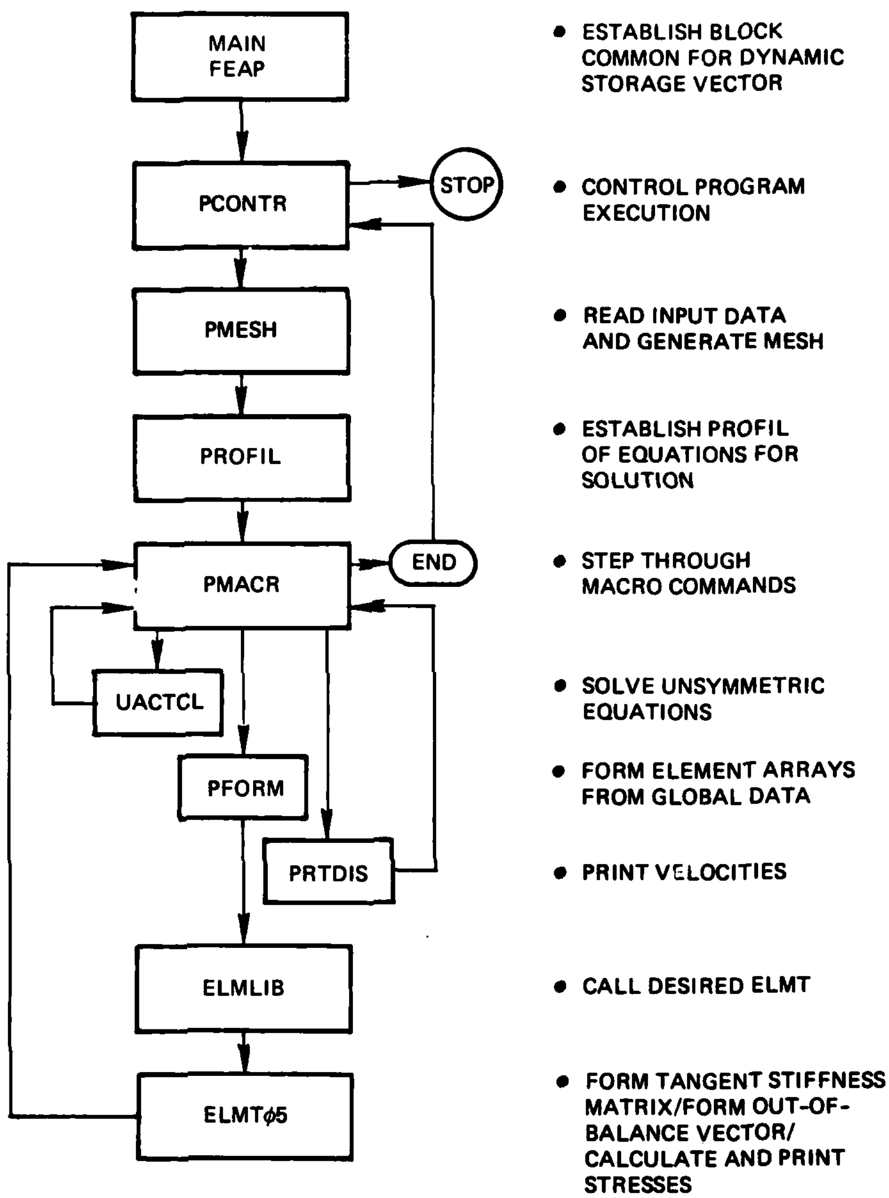


Figure 7 FEAP Flowchart

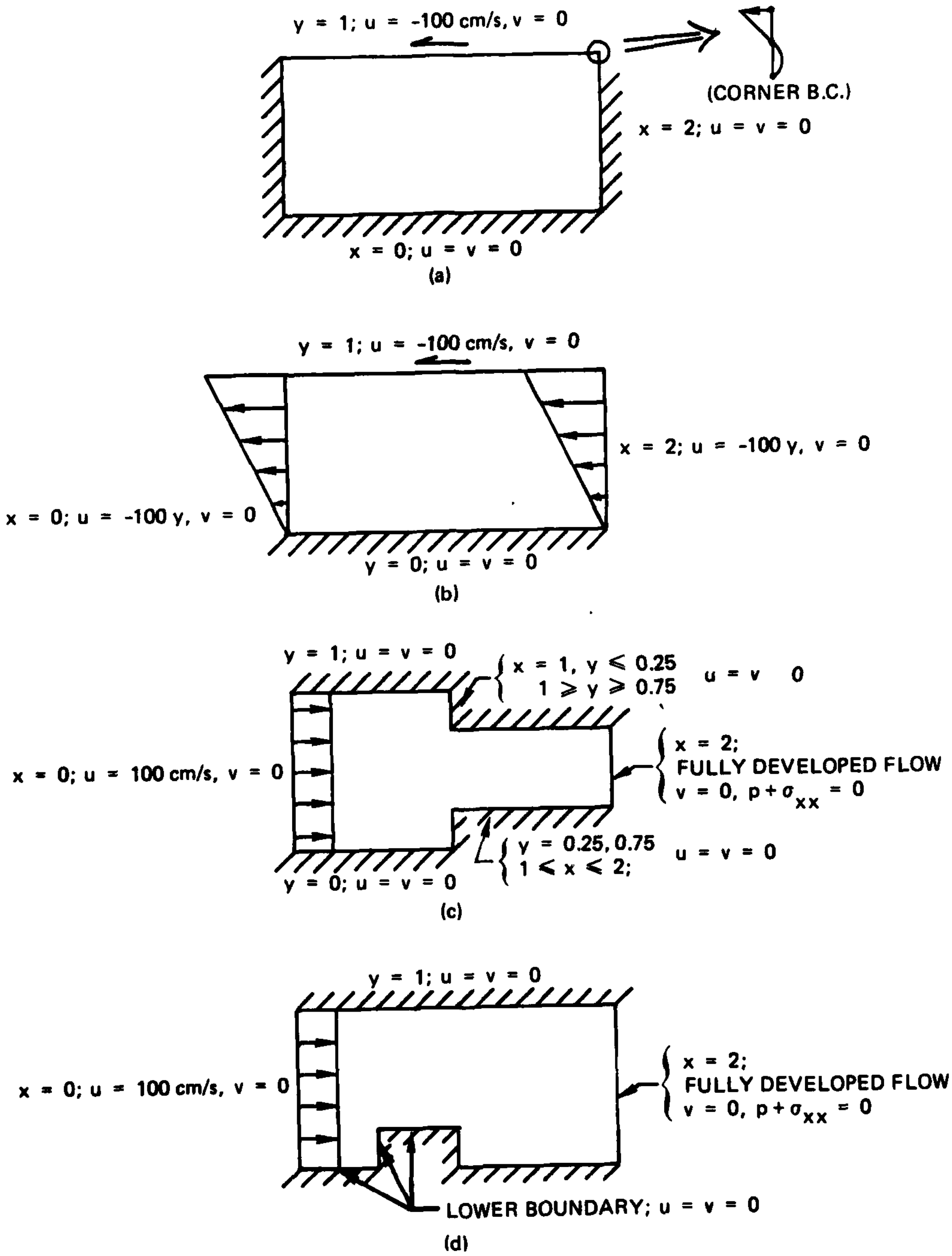


Figure 8 Flow Geometries and Boundary Conditions: (a) Cross Channel Flow (b) Plane Couette Flow (c) Entry Flow (d) Step Flow

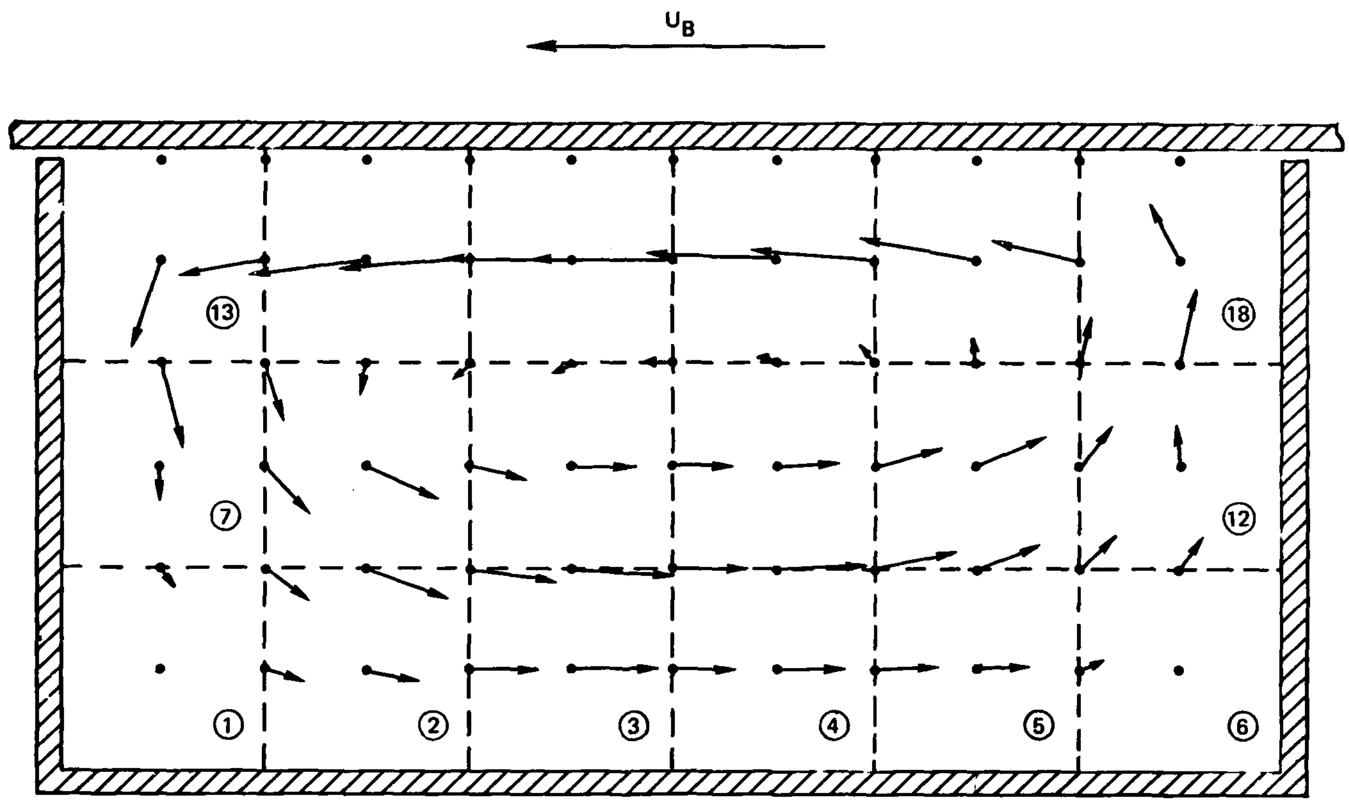


Figure 9 Velocity Flow Field for Linear Cross Channel Flow (Vectors scaled relative to U_B=100cm/sec)

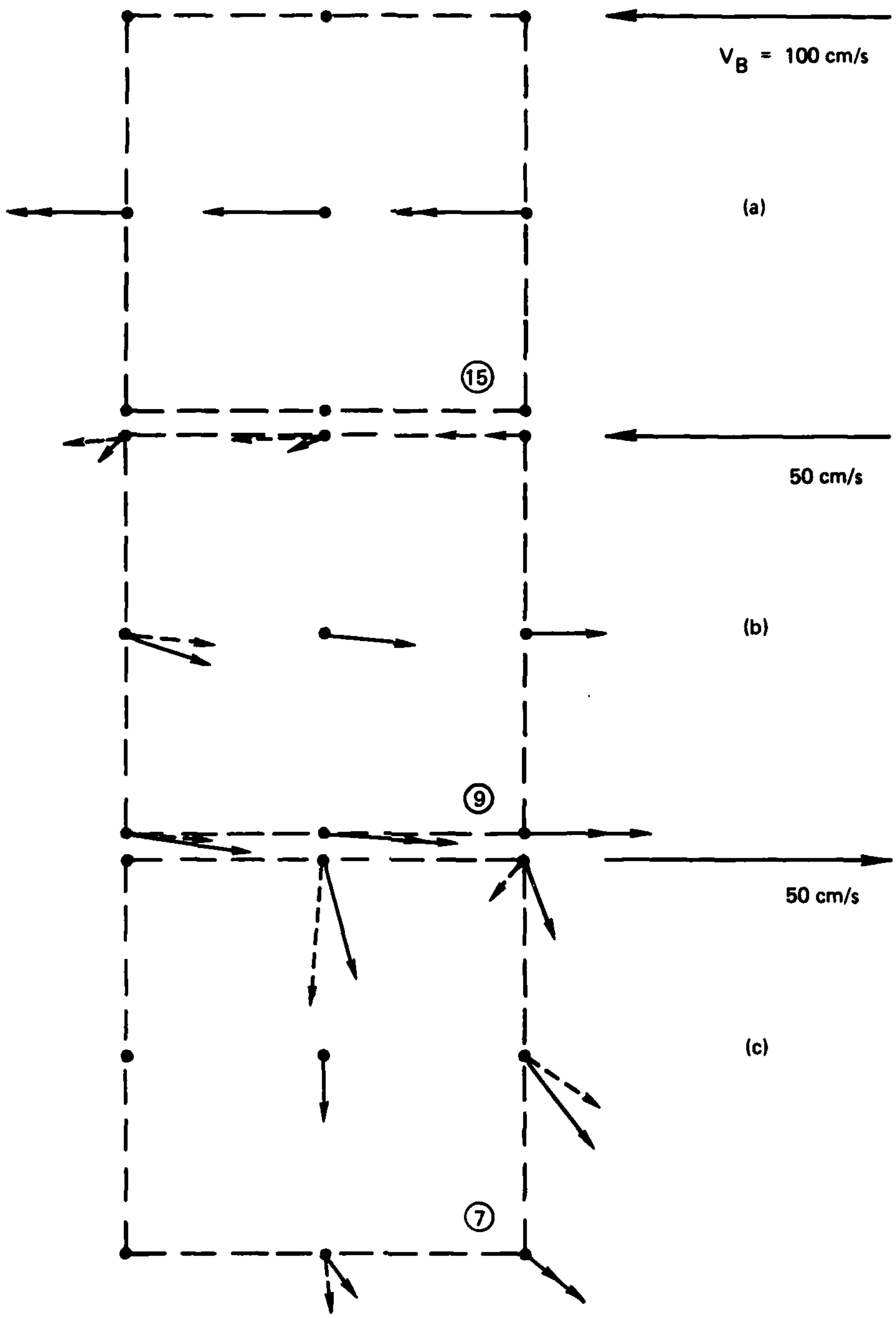


Figure 10 Velocity Comparisons of 9 and 8 Node Elements:
(a) Element 15 (b) Element 9 (c) Element 7
(Dashed Arrows are 8 Node Elements)

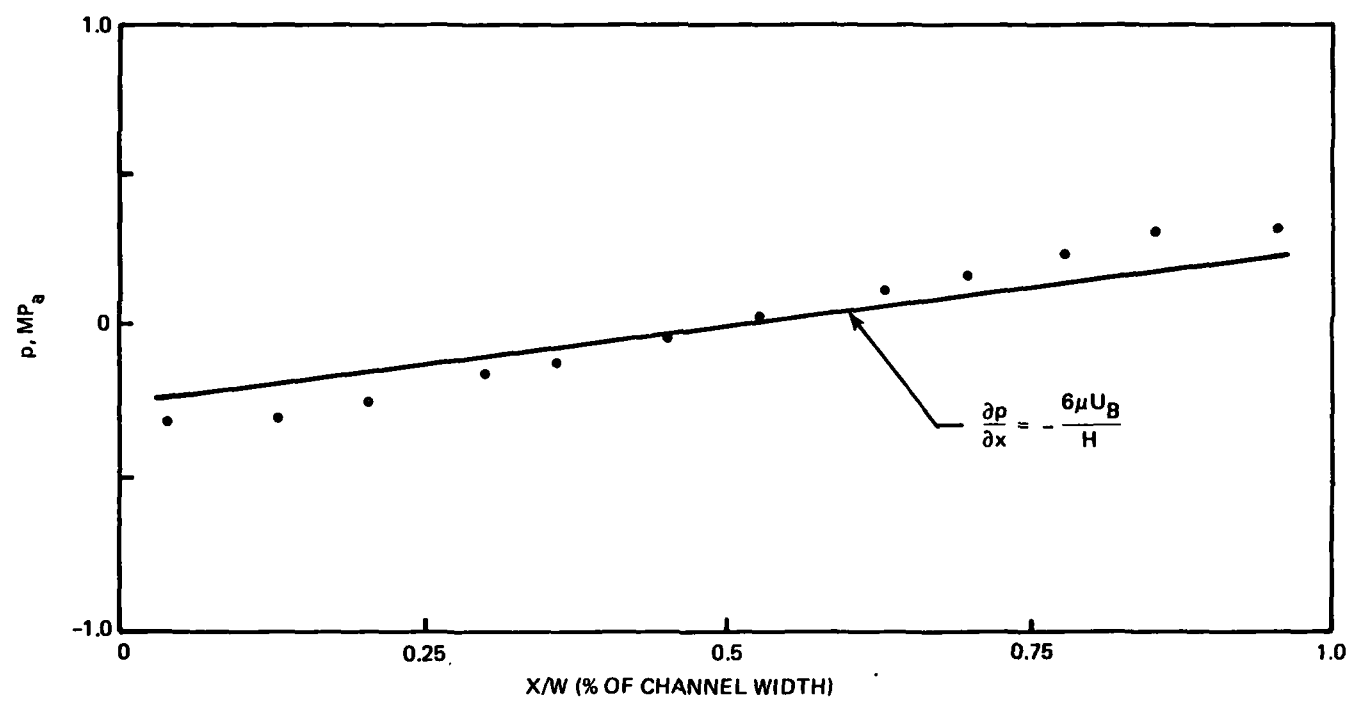


Figure 11 Computed Pressures in Cross Channel Flow Field (Line is Lubrication approximation; μ=viscosity, U_B=barrel velocity, H=channel height)

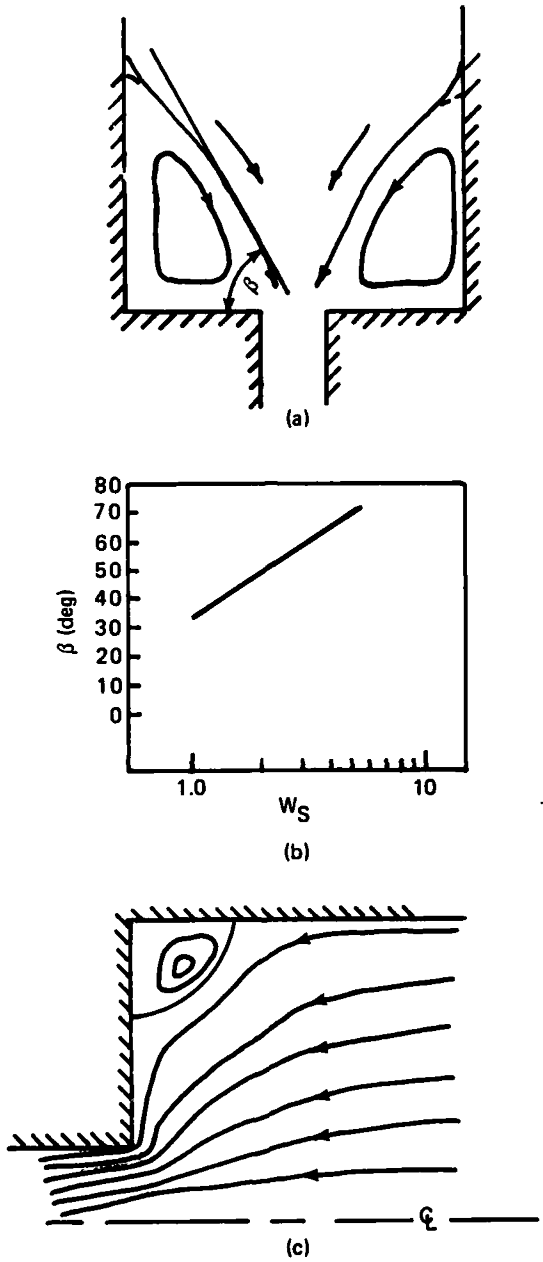


Figure 12 Fully Developed Flow Behavior of Viscoelastic Fluid Entering and Leaving a Contracting Channel: (a) Vortex angle β (after White [33]) (b) β vs Ws (after White [33]) (c) Finite Difference Calculation for Ws=0.6 (after Perera [27])

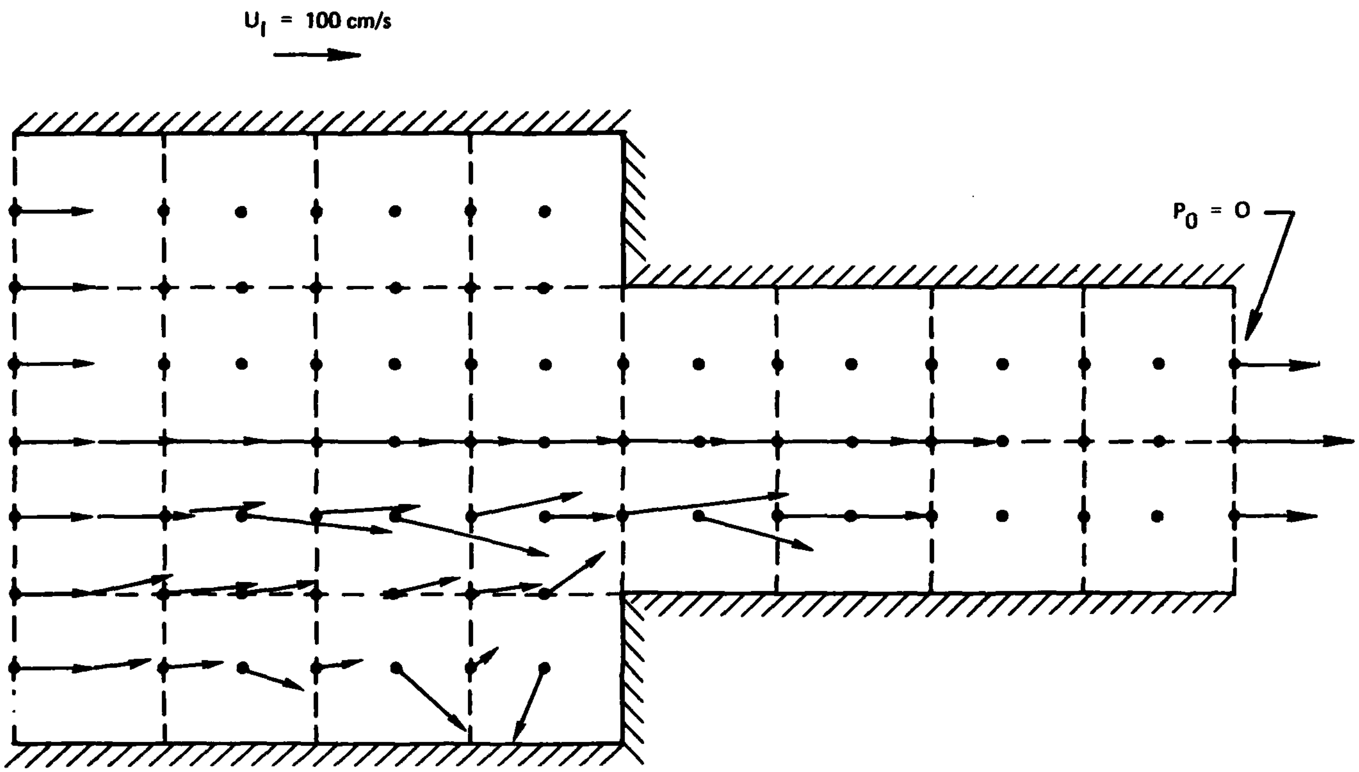


Figure 13 Uniform Inlet Velocity Flow Behavior of Newtonian Fluid Entering and Leaving a Contracting Channel (Flow is symmetric; Velocity vectors are scaled to the inlet U_I=100 cm/sec)

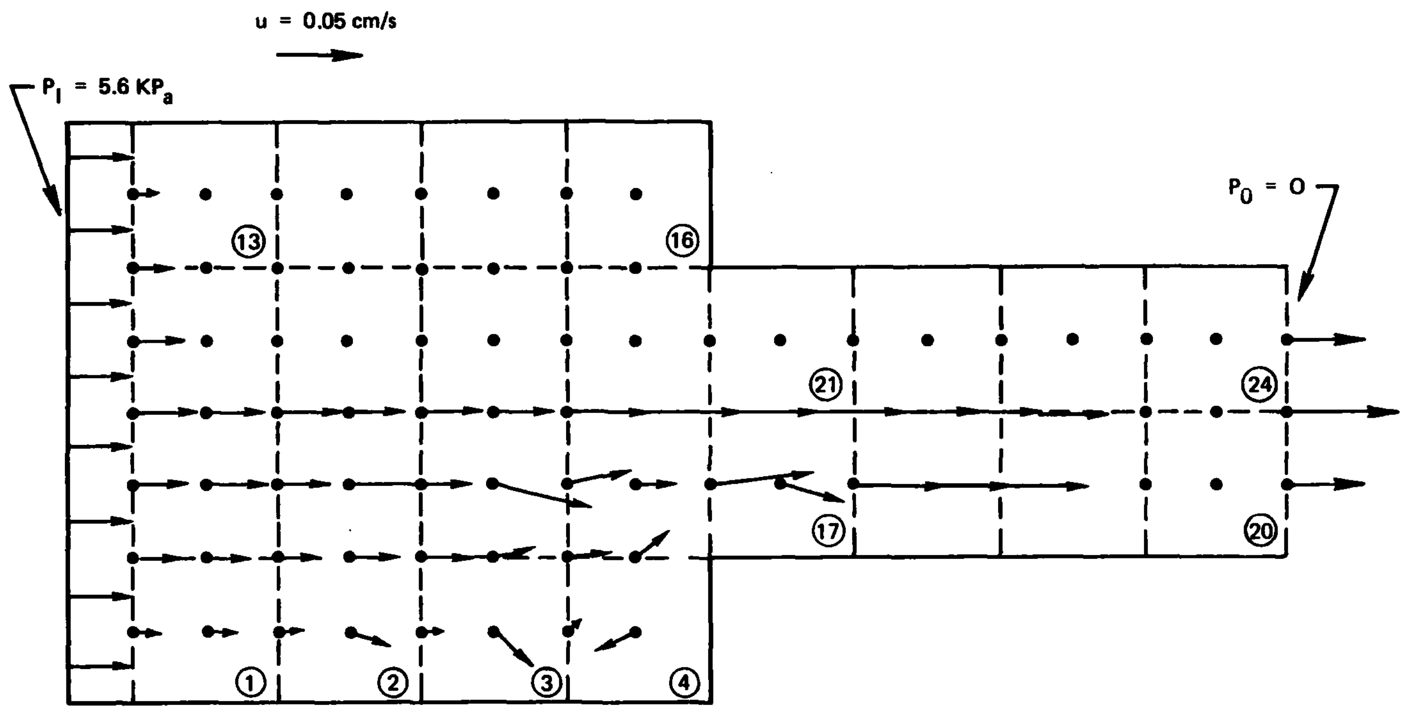


Figure 14 Fully Developed Flow Behavior of Newtonian Fluid Entering and Leaving a Contracting Channel (Flow is symmetric; Velocity vectors are scaled to u=0.05 cm/sec)

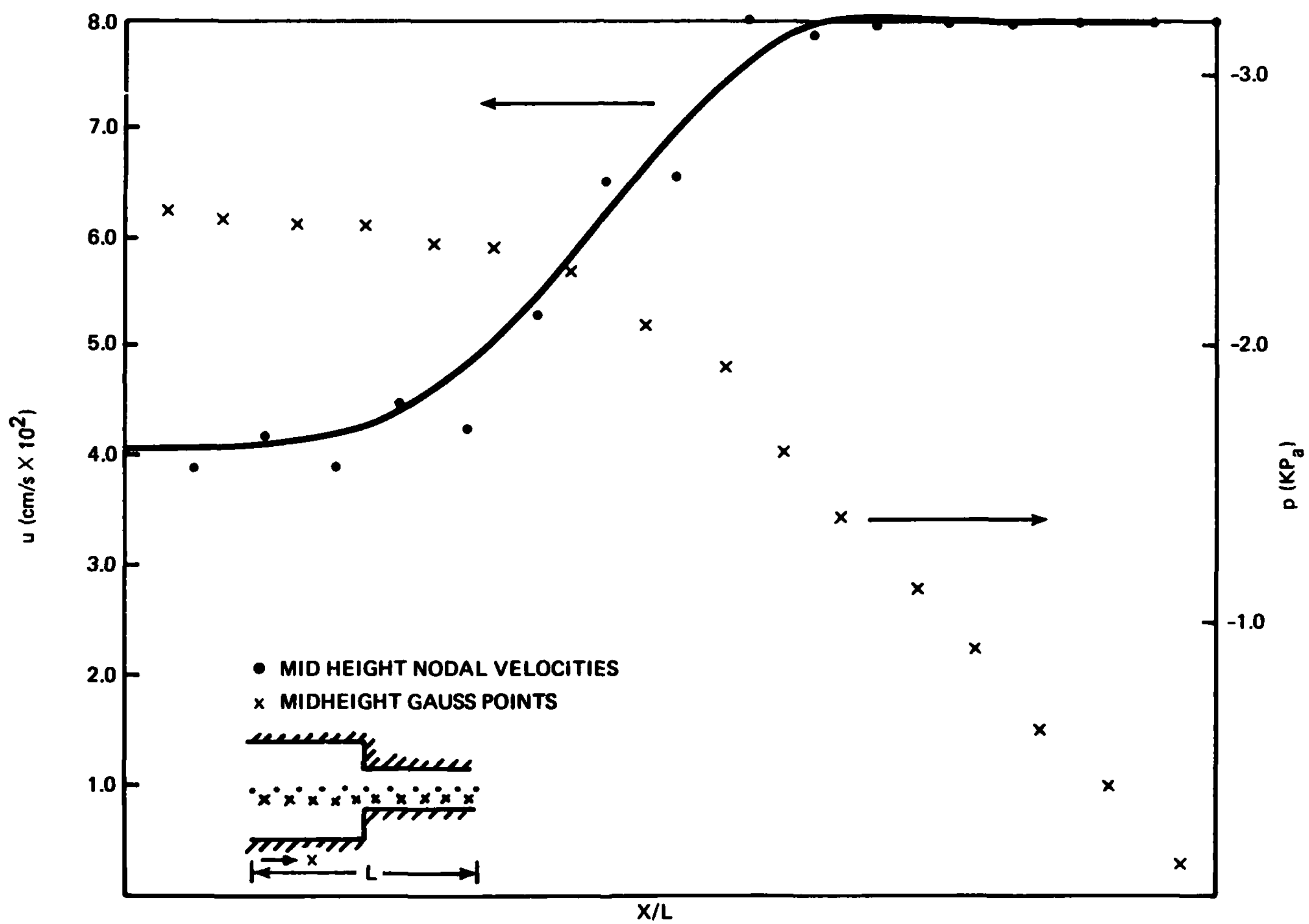


Figure 15 Velocities and Pressures for Newtonian Entry Flow (Curve for velocities is schematic only)

- TIME
  - STEADY
  - UNSTEADY
- TEMPERATURE
  - ISOTHERMAL
  - ADIABATIC
  - NONISOTHERMAL
- FLOW RATE
  - CREEPING
  - CONVECTION
- MATERIAL MODEL
  - NEWTONIAN
  - POWER LAW VISCOUS
  - VISCOELASTIC
- GEOMETRY
  - SINGLE REGION
  - CLOSED BOUNDARIES
  - PLANE FLOW
  - AXISYMMETRIC FLOW
  - 3D FLOW
  - FREE SURFACE
  - EXTRUDER/SPRUE RUNNER/GATE/MOLD
  - MULTIPLE GATE

Figure 16 Elements of a Complete Injection Molding Flow Analysis (Boxed items were evaluated in this study)

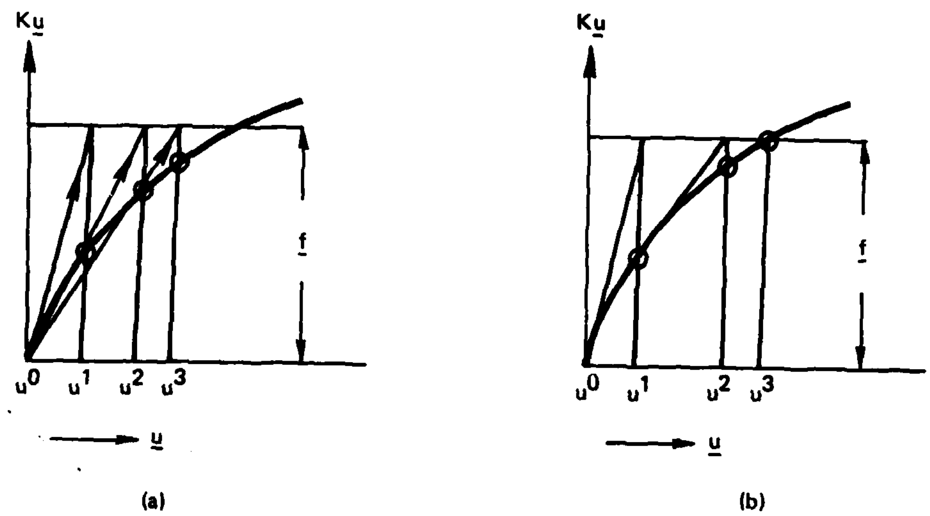
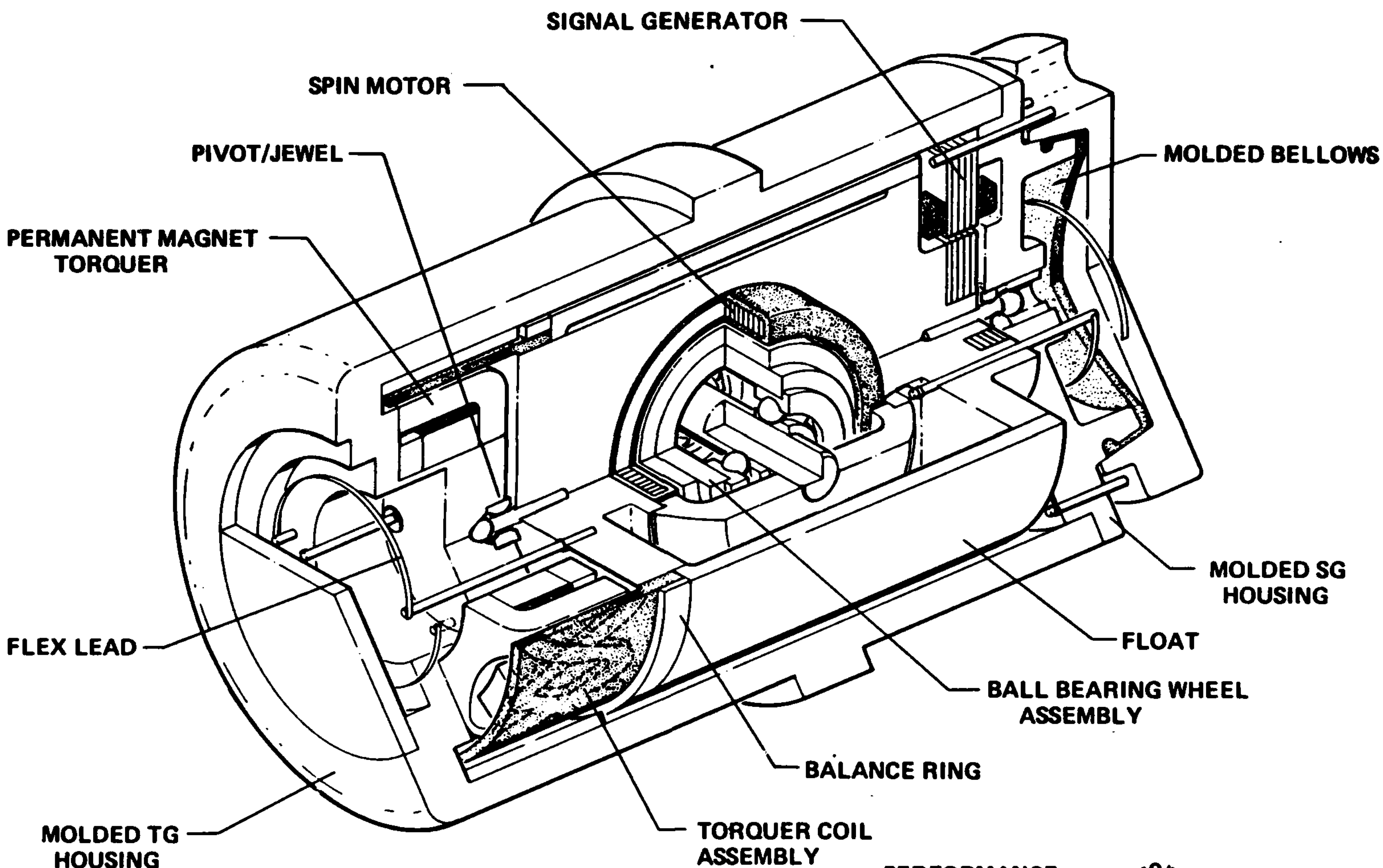


Figure 17 Solutions to Nonlinear Equations (a) Picard Iteration (b) Newton-Raphson Iteration

LOW COST GYRO/BASELINE 0 — LCG-101



PERFORMANCE — 1^{o}/hour
GYRO COST — \$600/unit

Figure 18 Cutaway of Gyroscope

RUN NO.	GEOMETRY	TYPE	CONVERGENCE (YES OR NO)	COST ($)
1	CROSS CHANNEL	LINEAR 18-9 NODE ELEM.	–	2.00
2	CROSS CHANNEL	LINEAR 18-8 NODE ELEM.	–	2.00
3	CROSS CHANNEL	LINEAR 72-8 NODE ELEM.	–	3.00
4	CROSS CHANNEL	CONVECTION (Re = 0.4) 18-9 NODE ELEM.	YES	3.50
5	CROSS CHANNEL	VISCOELASTIC (WS = 0.1) 18-9 NODE ELEM.	NO	10.00
6	CROSS CHANNEL	VISCOELASTIC (WS = 0.02) 18-9 NODE ELEM.	YES	21.35
7	CROSS CHANNEL	VISCOELASTIC (WS = 0.06) 18-9 NODE ELEM.	NO	32.96
8	PLANE COUETTE	LINEAR 18-9 NODE ELEM.	–	2.00
9	PLANE COUETTE	VISCOELASTIC (WS = 0.06) 18-9 NODE ELEM.	NO	12.47
10	PLANE COUETTE	VISCOELASTIC (WS = 0.02) 18-9 NODE ELEM.	YES	23.41
11	ENTRY	LINEAR 24-9 NODE ELEM.	–	2.00
12	ENTRY	VISCOELASTIC (WS = 0.01) 24-9 NODE ELEM.	TENDING AT 30 ITERATIONS	87.77
13	ENTRY	VISCOELASTIC (WS = 0.001) 24-9 NODE ELEM.	YES	77.00
14	ENTRY	VISCOELASTIC (WS = 0.03) 24-9 NODE ELEM.	NO	37.39
15	STEP	LINEAR 30-9 NODE ELEM.	–	2.50
16	STEP	CONVECTION (Re = 0.4) 30-9 NODE ELEM.	YES	21.46
17	STEP	VISCOELASTIC (WS = 0.01) 30-9 NODE ELEM.	YES	115.00
18	STEP	VISCOELASTIC (WS = 0.001) 30-9 NODE ELEM.	YES	79.12
19	STEP	VISCOELASTIC (WS = 0.03) 30-9 NODE ELEM.	NO	50.71

Table Computer Run Matrix

APPENDIX 1

Derivation of Elastic Stress Gradient Expressions

From figure 3 we can write the Taylor series approximations for $\nabla\sigma$ as:

Forward Difference:

$$\sigma^{i+1,j} = \sigma^{i,j} + \frac{\partial\sigma}{\partial x}\Big|_{i,j}\Delta x_f + \frac{\partial\sigma}{\partial y}\Big|_{i,j}\Delta y_f + \frac{1}{2}\,\frac{\partial^2\sigma}{\partial x^2}\Big|_{i,j}\Delta x_f^2 + \frac{1}{2}\,\frac{\partial^2\sigma}{\partial y^2}\Big|_{i,j}\Delta y_f^2 + \ldots$$

$$\sigma^{i,j+1} = \sigma^{i,j} + \frac{\partial\sigma}{\partial x}\Big|_{i,j}\Delta x_f^* + \frac{\partial\sigma}{\partial y}\Big|_{i,j}\Delta y_f^* + \frac{1}{2}\,\frac{\partial^2\sigma}{\partial x^2}\Big|_{i,j}\Delta x_f^{*2} + \frac{1}{2}\,\frac{\partial^2\sigma}{\partial y^2}\Big|_{i,j}\Delta y_f^{*2} + \ldots$$

Backward Difference:

$$\sigma^{i-1,j} = \sigma^{i,j} - \frac{\partial\sigma}{\partial x}\Big|_{i,j}\Delta x_b - \frac{\partial\sigma}{\partial y}\Big|_{i,j}\Delta y_b + \frac{1}{2}\,\frac{\partial^2\sigma}{\partial x^2}\Big|_{i,j}\Delta x_b^2 + \frac{1}{2}\,\frac{\partial^2\sigma}{\partial y^2}\Big|_{i,j}\Delta y_b^2 + \ldots$$

$$\sigma^{i,j-1} = \sigma^{i,j} - \frac{\partial\sigma}{\partial x}\Big|_{i,j}\Delta x_b^* - \frac{\partial\sigma}{\partial y}\Big|_{i,j}\Delta y_b^* + \frac{\partial^2\sigma}{\partial x^2}\Big|_{i,j}\Delta x_b^{*2} + \frac{1}{2}\,\frac{\partial^2\sigma}{\partial y^2}\Big|_{i,j}\Delta y_b^{*2} + \ldots$$

where:

$$\Delta x_f = x^{i+1,j} - x^{i,j}\ ,\quad \Delta y_f = y^{i+1,j} - y^{i,j}\ ,$$

$$\Delta x_f^* = x^{i,j+1} - x^{i,j}\ ,\quad \Delta y_f^* = y^{i,j+1} - y^{i,j}\ ,$$

$$\Delta x_b = x^{i,j} - x^{i-1,j}\ ,\quad \Delta y_b = y^{i,j} - y^{i-1,j}\ ,$$

and

$$\Delta x_b^* = x^{i,j} - x^{i,j-1} \;,\; \Delta y_b^* = y^{i,j} - y^{i,j-1}$$

Subtracting the first and second equations of the backward differences from the respective forward differences:

$$\sigma^{i+1,j} - \sigma^{i-1,j} = \frac{\partial\sigma}{\partial x}\Big|_{i,j}(\Delta x_f + \Delta x_b) + \frac{\partial\sigma}{\partial y}\Big|_{i,j}(\Delta y_f + \Delta y_b) + \frac{1}{2}\,\frac{\partial^2\sigma}{\partial x^2}\Big|_{i,j}(\Delta x_f^2 - \Delta x_b^2) + \frac{1}{2}\,\frac{\partial^2\sigma}{\partial y^2}\Big|_{i,j}(\Delta y_f^2 - \Delta y_b^2) + \ldots O(\Delta^3)$$

$$\sigma^{i,j+1} - \sigma^{i,j-1} = \frac{\partial\sigma}{\partial x}\Big|_{i,j}(\Delta x_f^* + \Delta x_b^*) + \frac{\partial\sigma}{\partial y}\Big|_{i,j}(\Delta y_f^* + \Delta y_b^*) + \frac{1}{2}\,\frac{\partial^2\sigma}{\partial x^2}\Big|_{i,j}(\Delta x_f^{*2} - \Delta x_b^{*2}) + \frac{1}{2}\,\frac{\partial^2\sigma}{\partial y^2}\Big|_{i,j}(\Delta y_f^{*2} - \Delta y_b^{*2}) + \ldots O(\Delta^3)$$

Assuming that all differences of the intervals squared are infinitesimal (zero for the uniform mesh case) and solving for the gradients we have in matrix form:

$$\begin{bmatrix} (\Delta x_f + \Delta x_b) & (\Delta y_f + \Delta y_b) \\ (\Delta x_f^* + \Delta x_b^*) & (\Delta y_f^* + \Delta y_b^*) \end{bmatrix} \begin{bmatrix} \frac{\partial\sigma}{\partial x} \\ \frac{\partial\sigma}{\partial y} \end{bmatrix} = \begin{bmatrix} \sigma^{i+1,j} - \sigma^{i-1,j} \\ \sigma^{i,j+1} - \sigma^{i,j-1} \end{bmatrix}$$

We can use Cramer's rule for the solution since the determinant of the coefficients of the gradients can never vanish. Therefore:

$$\frac{\partial\sigma}{\partial x} = \frac{(\sigma^{i+1,j} - \sigma^{i-1,j})(\Delta y_f^* + \Delta y_b^*) - (\sigma^{i,j+1} - \sigma^{i,j-1})(\Delta y_f + \Delta y_b)}{(\Delta x_f + \Delta x_b)(\Delta y_f^* + \Delta y_b^*) - (\Delta x_f^* + \Delta x_b^*)(\Delta y_f + \Delta y_b)}$$

$$\frac{\partial\sigma}{\partial y} = \frac{(\sigma^{i,j+1} - \sigma^{i,j-1})(\Delta x_f + \Delta x_b) - (\sigma^{i+1,j} - \sigma^{i-1,j})(\Delta x_f^* + \Delta x_b^*)}{(\Delta x_f + \Delta x_b)(\Delta y_f^* + \Delta y_b^*) - (\Delta x_f^* + \Delta x_b^*)(\Delta y_f + \Delta y_b)}$$

When substitutions are made for the Δ terms we obtain equations IV.19.

APPENDIX 2

Calculation of the Global Second Derivatives

A subroutine ESHAP was written to calculate the global second derivatives of the velocity vector. For a nine-node Lagragian isoparametric element the trial functions are:

$$N_1 = \frac{1}{4}(r^2 - r)(s^2 - s)$$

$$N_2 = \frac{1}{4}(r^2 + r)(s^2 - s)$$

$$N_3 = \frac{1}{4}(r^2 + r)(s^2 + s)$$

$$N_4 = \frac{1}{4}(r^2 - r)(s^2 + s)$$

$$N_5 = -\frac{1}{2}(r^2 - 1)(s^2 - s)$$

$$N_6 = -\frac{1}{2}(r^2 + r)(s^2 - 1)$$

$$N_7 = -\frac{1}{2}(r^2 - 1)(s^2 + s)$$

$$N_8 = -\frac{1}{2}(r^2 - r)(s^2 - 1)$$

$$N_9 = (r^2 - 1)(s^2 - 1)$$

We can form the following table:

AD-A106 740 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH F/G 20/11
A FINITE ELEMENT MODEL OF A WHITE-METZNER VISCOELASTIC POLYMER --ETC(U)
FEB 81 B R COLLINS
UNCLASSIFIED AFIT-CI-81-31T NL

2 of 2
AD
A106740

END
DATE
FILMED
11-81
DTIC

	$\frac{\partial^2}{\partial r^2}$	$\frac{\partial^2}{\partial s^2}$	$\frac{\partial^2}{\partial r \partial s}$
N_1	$\frac{1}{2}(s^2 - s)$	$\frac{1}{2}(r^2 - r)$	$\frac{1}{4}(2r - 1)(2s - 1)$
N_2	$\frac{1}{2}(s^2 - s)$	$\frac{1}{2}(r^2 + r)$	$\frac{1}{4}(2r + 1)(2s - 1)$
N_3	$\frac{1}{2}(s^2 + s)$	$\frac{1}{2}(r^2 + r)$	$\frac{1}{4}(2r + 1)(2s + 1)$
N_4	$\frac{1}{2}(s^2 + s)$	$\frac{1}{2}(r^2 - r)$	$\frac{1}{4}(2r - 1)(2s + 1)$
N_5	$s - s^2$	$1 - r^2$	$r(1 - 2s)$
N_6	$1 - s^2$	$-(r^2 + r)$	$-s(2r + 1)$
N_7	$-(s^2 + s)$	$1 - r$	$-r(2s + 1)$
N_8	$1 - s^2$	$r - r^2$	$s(1 - 2r)$
N_9	$2(s^2 - 1)$	$2(r^2 - 1)$	$4rs$

Writing the expressions for the second derivatives we have:

$$\frac{\partial^2}{\partial_r^2} = \frac{\partial}{\partial r}\left[\frac{\partial x}{\partial r}\frac{\partial}{\partial x} + \frac{\partial y}{\partial r}\frac{\partial}{\partial y}\right], \quad \frac{\partial^2}{\partial s^2} = \frac{\partial}{\partial s}\left[\frac{\partial x}{\partial s}\frac{\partial}{\partial x} + \frac{\partial y}{\partial s}\frac{\partial}{\partial y}\right],$$

$$\frac{\partial^2}{\partial r \partial s} = \frac{\partial}{\partial s}\left[\frac{\partial x}{\partial r}\frac{\partial}{\partial x} + \frac{\partial y}{\partial r}\frac{\partial}{\partial y}\right]$$

where r,s are local coordinates and x,y are global coordinates and the terms in brackets are merely the chain rules for forming the coordinate transformations (e.g., $\frac{\partial}{\partial r} = \frac{\partial x}{\partial r}\frac{\partial}{\partial x} + \frac{\partial y}{\partial r}\frac{\partial}{\partial y}$)

Recognizing that terms such as $\frac{\partial^2 x}{\partial r \partial x}$ and $\frac{\partial^2 y}{\partial s \partial x}$ are zero, we can write the transformations in matrix form as:

$$\begin{bmatrix} \frac{\partial^2}{\partial r^2} \\ \\ \frac{\partial^2}{\partial s^2} \\ \\ \frac{\partial^2}{\partial r \partial s} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial x}{\partial r}\right)^2 & \left(\frac{\partial y}{\partial r}\right)^2 & 2\frac{\partial x}{\partial r}\frac{\partial y}{\partial r} \\ \\ \left(\frac{\partial x}{\partial s}\right)^2 & \left(\frac{\partial y}{\partial s}\right)^2 & 2\frac{\partial x}{\partial s}\frac{\partial y}{\partial s} \\ \\ \frac{\partial x}{\partial r}\frac{\partial x}{\partial s} & \frac{\partial y}{\partial r}\frac{\partial y}{\partial s} & \left(\frac{\partial x}{\partial r}\frac{\partial y}{\partial s} + \frac{\partial y}{\partial r}\frac{\partial x}{\partial s}\right) \end{bmatrix} \begin{bmatrix} \frac{\partial^2}{\partial x^2} \\ \\ \frac{\partial^2}{\partial y^2} \\ \\ \frac{\partial^2}{\partial x \partial y} \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{\partial^2 x}{\partial r^2} & \frac{\partial^2 y}{\partial r^2} \\ \\ \frac{\partial^2 x}{\partial s^2} & \frac{\partial^2 y}{\partial s^2} \\ \\ \frac{\partial^2 x}{\partial r \partial s} & \frac{\partial^2 y}{\partial r \partial s} \end{bmatrix} \underline{J}^{-1} \begin{bmatrix} \frac{\partial}{\partial r} \\ \\ \frac{\partial}{\partial s} \end{bmatrix}$$

Where the Jacobian

$$\underline{J} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial s} \\ \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial s} \end{bmatrix}$$

has been used. All the terms in this equation are available at the Gauss points e.g.

$$\left.\frac{\partial^2 x}{\partial r^2}\right|_{G.P.} = \sum_{i=1}^{9} \left.\frac{\partial^2 Ni}{\partial r^2}\right|_{G.P.} X_i$$

where X_i are the x coordinates of node i.

We can then solve for the global second derivatives according to:

$$
\begin{bmatrix} \dfrac{\partial^2}{\partial x^2} \\[2ex] \dfrac{\partial^2}{\partial y^2} \\[2ex] \dfrac{\partial^2}{\partial x \partial y} \end{bmatrix}
=
\begin{bmatrix}
\left(\dfrac{\partial x}{\partial r}\right)^2 & \left(\dfrac{\partial y}{\partial r}\right)^2 & 2\dfrac{\partial x}{\partial r}\,\dfrac{\partial y}{\partial r} \\[2ex]
\left(\dfrac{\partial x}{\partial s}\right)^2 & \left(\dfrac{\partial y}{\partial s}\right)^2 & 2\dfrac{\partial x}{\partial s}\,\dfrac{\partial y}{\partial s} \\[2ex]
\dfrac{\partial x}{\partial r}\,\dfrac{\partial x}{\partial s} & \dfrac{\partial y}{\partial r}\,\dfrac{\partial y}{\partial s} & \left(\dfrac{\partial x}{\partial r}\,\dfrac{\partial y}{\partial s} + \dfrac{\partial y}{\partial r}\,\dfrac{\partial x}{\partial s}\right)
\end{bmatrix}^{-1}
\left\{
\begin{bmatrix} \dfrac{\partial^2}{\partial r^2} \\[2ex] \dfrac{\partial^2}{\partial s^2} \\[2ex] \dfrac{\partial^2}{\partial r \partial s} \end{bmatrix}
-
\begin{bmatrix}
\dfrac{\partial^2 x}{\partial r^2} & \dfrac{\partial^2 y}{\partial r^2} \\[2ex]
\dfrac{\partial^2 x}{\partial s^2} & \dfrac{\partial^2 y}{\partial s^2} \\[2ex]
\dfrac{\partial^2 x}{\partial r \partial s} & \dfrac{\partial^2 y}{\partial r \partial s}
\end{bmatrix}
\underline{J}^{-1}
\begin{bmatrix} \dfrac{\partial}{\partial r} \\[2ex] \dfrac{\partial}{\partial s} \end{bmatrix}
\right\}
$$

A value is therefore returned for each of the nine trial functions for the three global second derivatives.

APPENDIX 3

Listing of New Subroutines

1. ELMTØ5
2. ELMTØ6
3. ESHAP
4. PFORM
5. CMATRX
6. FPSIG

BRC4066 (FOREGROUND): OUTPUT FROM TSO XPRINT

AT 18:02:48 ON 12/07/80 - BRC4066.ELMT05.FORT

```
      SUBROUTINE ELMT05( D , UL , XL , IX , TL , S , P ,NDF,NDM,NST,ISW)
C
C**********************                 ********************************
C**********************     ELMT05      ********************************
C**********************                 ********************************
C
C     A GENERAL PENALTY ELEMENT FOR INCOMPRESSIBLE FLUID FLOW
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 XMT(4,6)/8*1.D0,2*0.D0,2*1.D0,12*0.D0/,
     1 XM(6,4)/2*1.D0,4*0.D0,2*1.D0,4*0.D0,3*1.D0,3*0.D0,3*1.D0,3*0.D0/
      INTEGER LB(4)/3,3,4,6/
      COMMON /CDATA/O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      COMMON /ELDATA/DM,N,MA,MOT,IEL,NEL
      COMMON /FVISC/K2
      COMMON /TAYLR/ESIG1(4,2,50),ESIG2(4,2,50),ESIG3(4,2,50),YY(4,2,50
     1 ),ELAS1(4,2,50),ELAS2(4,2,50),ELAS3(4,2,50),EOSIG(4,3,50)
      DIMENSION D(30),UL(NDF,1),XL(NDM,1),IX(1),TL(1),
     1   S(NST,1),P(1),SHP(3,9),SG(9),TG(9),WG(9),
     2   SIG(7),EPS(6),BSIG(3),XX(3),B(18),DB(6,3),BTDB(3,3),
     3   BU(6),XMTB(3),XMTBT(3),PEN(3,3),DU(3),DLTEE(3),
     4   V(2)  ,DV(2,2)    ,XN(2,2)    ,ADVEC(2,2)    ,CADVEC(2,2)
     5  ,AITER(3,3),ESHP(3,9),DDV(3,2)
      DATA PI/3.1415926536D0/
C
      IF (ISW.EQ.1) GO TO 1
      ITYPE = D(30)
      L = D(28)
      RHO = D(27)
      XLAM = D(26)
      XMU = D(25)
      XK = D(24)
      C = D(23)
      N1 = D(20)
      HEAT = D(19)
      LLB = LB(ITYPE)
      K2 = D(18)
      N3 = D(17)
      G = D(16)
      P4 = D(15)
C
C     BRANCH TO CORRECT ARRAY PROCESSOR
C
      GO TO (1,2,3,3,5,3,3),ISW
C
C
C
C     ISW = 1:   READ MATERIAL PROPERTIES, DEVELOP
C                DIAGONAL-STORAGE D MATRIX
C
C
    1 CALL DFMTRX(D)
      LINT = 0
C
      RETURN
C
```

```
    2 RETURN
C
C
C
C     ISW = 3:   FORM ELEMENT STIFFNESS MATRIX
C
C
C
    3 CONTINUE
C
C     LOOP OVER GAUSS INTEGRATION POINTS
C     COMPUTE UNSYMMETRIC STIFFNESS MATRIX
C
      IF (L**NDM .NE. LINT) CALL PGAUSS (L,LINT,SG,TG,WG)
      DO 33 LL=1,LINT
C
      CALL SHAPE (SG(LL),TG(LL),XL,SHP,XSJ,NDM,NEL,IX,.FALSE.)
      WGT=XSJ*WG(LL)
C
C     COMPUTE RADIUS FOR AXISYMMETRIC CASE
C
      IF (ITYPE.NE.3) GO TO 302
         RR=0.D0
         DO 301 I=1,NEL
            RR=RR+SHP(3,I)*XL(1,I)
301   CONTINUE
      WGT=WGT*2.D0*PI*RR
302   CONTINUE
C
C     COMPUTE COORDINATES, VELOCITIES AND GRADIENTS FOR CONVECTIVE TERM
C
      DO 32 I=1,NDM
         XX(I)=0.D0
         V(I)=0.D0
      DO 31 K=1,NEL
         XX(I)=XX(I)+SHP(3,K)*XL(I,K)
         V(I)=V(I)+SHP(3,K)*UL(I,K)
31    CONTINUE
      YY(LL,I,N) = XX(I)
      DO 32 J=1,NDM
      DV(I,J)=0.D0
      DO 32 K=1,NEL
32    DV(I,J)=DV(I,J)+SHP(J,K)*UL(I,K)
C
C     COMPUTE NONLINEAR VISCOSITY CORRECTION
C
      XNLNR=1.D0
      IF (P4.EQ.1.) GO TO 325
      A1=2.D0*(DV(1,1)**2+DV(2,2)**2)+(DV(1,2)+DV(2,1))**2
      IF (ITYPE.NE.3) GO TO 320
      A1=A1+2.D0*(V(1)/XX(1))**2
320   XNLNR=XNLNR/(1.D0+A1**((1.D0-P4)/2.D0))
325   VISLAM = 0.D0
      IF (G.EQ.0.D0) GO TO 9
      VISLAM = XNLNR*XMU/G+VISLAM
9     IF (ISW.EQ.6.OR.ISW.EQ.4.OR.ISW.EQ.7) GO TO 47
C     LOOP OVER COLUMNS, FORMING DB, MT*B, AND (DEL.(NU)T)T*N
      DO 46 J=1,NEL
C
         CALL BMATRX(B,J,ITYPE,SHP,RR)
```

```
         CALL VMULDF(D,LLB,B,NDM,LLB,DB,6)
         CALL VMULFF(XMT(ITYPE,1),B,1,LLB,NDM,4,LLB,XMTB,1,IER)
         DO 37 IDEX=1,NDM
         DO 37 JDEX=1,NDM
         IF (IDEX.EQ.JDEX) XN(IDEX,JDEX)=SHP(3,J)
 37      IF (IDEX.NE.JDEX) XN(IDEX,JDEX)=0.D0
         CALL VMULFF(DV,XN,NDM,NDM,NDM,NDM,NDM,ADVEC,NDM,IER)
      JJ=(J-1)*NDF+1
C
C     LOOP OVER ROWS, FORMING BT*(DB),(MTB)T*MTB, AND NT(DEL.(NU)T)T*N
C
      DO 45 I=1,NEL
C
         CALL BMATRX(B,I,ITYPE,SHP,RR)
         CALL VMULFM(B,DB,LLB,NDM,NDM,LLB,6,BTDB,3,IER)
         CALL VMULFF(XMT(ITYPE,1),B,1,LLB,NDM,4,LLB,XMTBT,1,IER)
         CALL VMULFM(XMTBT,XMTB,1,NDM,NDM,1,1,PEN,3,IER)
         CALL VMULFM(XN,ADVEC,NDM,NDM,NDM,NDM,NDM,CADVEC,NDM,IER)
      II=(I-1)*NDF+1
C
C     ADD TO ELEMENT STIFFNESS MATRIX S(NST,NST)
C
         CALL MXADD(S(II,JJ),NST,BTDB,3,NDM,NDM,WGT*XNLNR)
         CALL MXADD(S(II,JJ),NST,PEN,3,NDM,NDM,WGT*XLAM)
         CALL MXADD(S(II,JJ),NST,CADVEC,NDM,NDM,NDM,WGT*RHO)
C
C     ADD THERMAL STIFFNESS
C
      IF (N1.EQ.1) A2=XK*DOT(SHP(1,I),SHP(1,J),NDM)
      IF (N1.EQ.1) S(II+NDM,JJ+NDM)=S(II+NDM,JJ+NDM)+A2*WGT
 45   CONTINUE
 46   CONTINUE
      IF (ISW.EQ.3) GO TO 65
 47   CONTINUE
      IF (ISW.EQ.4.OR.ISW.EQ.6) GO TO 60
C
C     CALCULATE ESIG(LL,2,N):  ELASTIC STRESS AT K + 1 ITERATION
C
C     SET UP A MATRIX FOR PLANE FLOW
C
      AITER(1,1) = DV(1,1)*2.D0
      AITER(2,1) = 0.D0
      AITER(3,1) = DV(2,1)
      AITER(1,2) = 0.D0
      AITER(2,2) = DV(2,2)*2.D0
      AITER(3,2) = DV(1,2)
      AITER(1,3) = DV(1,2)*2.D0
      AITER(2,3) = DV(2,1)*2.D0
      AITER(3,3) = DV(1,1) + DV(2,2)
C
C     COMPUTE VISCOUS STRESSES AT GAUSS POINTS:  SIG = D*BU
C
 60   SIG(1) = XMU*2.D0*DV(1,1)*XNLNR
      SIG(2) = XMU*2.D0*DV(2,2)*XNLNR
      SIG(3) = XMU*(DV(1,2) + DV(2,1))*XNLNR
      SIG(7) = XLAM*(DV(1,1) + DV(2,2))
      IF(ITYPE.NE.3) GO TO 61
      SIG(4) = SIG(3)
      SIG(3) = (V(1)/XX(1))*XMU*XNLNR*2.D0
      SIG(7) = SIG(7) + XLAM*(V(1)/XX(1))
```

```
61    IF(ISW.EQ.4.CR.ISW.EQ.6) GO TO 62
C
C     CALCULATE VISCOUS STRESS GRADIENT (DEL(SIGMA))
C
      CALL ESHAP(SG(LL),TG(LL),XL,ESHP,NDM,NEL,IX)
C
C     FORM CONVECTION DERIVATIVE OF STRESS:  ONLY 2D FLOW
C
      DO 21 I=1,2
      DO 21 J=1,3
21    DDV(J,I) = 0.D0
      DO 22 K=1,NEL
      DDV(1,1) = DDV(1,1) + 2.D0*XMU*XNLNR*ESHP(1,K)*UL(1,K)
      DDV(1,2) = DDV(1,2) + 2.D0*XMU*XNLNR*ESHP(3,K)*UL(1,K)
      DDV(2,1) = DDV(2,1) + 2.D0*XMU*XNLNR*ESHP(3,K)*UL(2,K)
      DDV(2,2) = DDV(2,2) + 2.D0*XMU*XNLNR*ESHP(2,K)*UL(2,K)
      DDV(3,1) = DDV(3,1) + XMU*XNLNR*(ESHP(1,K)*UL(2,K)+ESHP(3,K)*UL(
     1          1,K))
      DDV(3,2) = DDV(3,2) + XMU*XNLNR*(ESHP(3,K)*UL(2,K)+ESHP(2,K)*UL(
     1          1,K))
22    CONTINUE
C
C     SOLVE ESIG(LL,2,N):  ONLY 2D FLOW
C
      ESIG1(LL,2,N) = VISLAM*((AITER(1,1)*(SIG(1)+ESIG1(LL,1,N))+
     1 AITER(1,2)*(SIG(2)+ESIG2(LL,1,N))+AITER(1,3)*(SIG(3)+ESIG3(
     2 LL,1,N)))-(V(1)*(DDV(1,1)+ELAS1(LL,1,N))+V(2)*(DDV(1,2)+ELAS1(
     3 LL,2,N))))
      ESIG2(LL,2,N) = VISLAM*((AITER(2,1)*(SIG(1)+ESIG1(LL,1,N))+
     1 AITER(2,2)*(SIG(2)+ESIG2(LL,1,N))+AITER(2,3)*(SIG(3)+ESIG3(
     2 LL,1,N)))-(V(1)*(DDV(2,1)+ELAS2(LL,1,N))+V(2)*(DDV(2,2)+ELAS2(
     3 LL,2,N))))
      ESIG3(LL,2,N) = VISLAM*((AITER(3,1)*(SIG(1)+ESIG1(LL,1,N))+
     1 AITER(3,2)*(SIG(2)+ESIG2(LL,1,N))+AITER(3,3)*(SIG(3)+ESIG3(
     2 LL,1,N)))-(V(1)*(DDV(3,1)+ELAS3(LL,1,N))+V(2)*(DDV(3,2)+ELAS3(
     3 LL,2,N))))
C
C     UPDATE BOUNDARY STRESSES BOSIG(NODE,DIRECTION,ELMT. NO.)
C
      IF (G.EQ.0.D0) GO TO 65
      BOSIG(LL,1,N) = ESIG1(LL,2,N) + ELAS1(LL,1,N)*(XL(1,LL)-
     1               YY(LL,1,N)) + ELAS1(LL,2,N)*(XL(2,LL)-YY(LL,2,N))
      BOSIG(LL,2,N) = ESIG2(LL,2,N) + ELAS2(LL,1,N)*(XL(1,LL)-
     1               YY(LL,1,N)) + ELAS2(LL,2,N)*(XL(2,LL)-YY(LL,2,N))
      BOSIG(LL,3,N) = ESIG3(LL,2,N) + ELAS3(LL,1,N)*(XL(1,LL)-
     1               YY(LL,1,N)) + ELAS3(LL,2,N)*(XL(2,LL)-YY(LL,2,N))
      GO TO 65
C
C     PRINT STRESSES IF ISW=4, OTHERWISE BRANCH TO COMPUTE
C     UNBALANCED FCRCE VECTOR
C
62    IF (ISW.EQ.6) GO TO 66
      XMAX = DMAX1(DABS(XL(1,4)-XL(1,1)),DABS(XL(1,3)-XL(1,2)),
     1 DABS(XL(2,4)-XL(2,1)),DABS(XL(2,3)-XL(2,2)))
      SIG(5) = (RHO/(XMU*XNLNR))*DSQRT(V(1)**2+V(2)**2)*XMAX
      SIG(6) = .ISLAM*DSQRT(V(1)**2+V(2)**2)/XMAX
      CALL  FPSIG(XX,ESIG1(LL,2,N),ESIG2(LL,2,N),ESIG3(LL,2,N),SIG,
     1 ITYPE,NDF)
      GO TO 65
C
```

```
C     LOOP OVER NODES TO COMPUTE UNBALANCED FORCE VECTOR:                00002207
C     P = P1 - BT*SIG - NT*ELAS(LL,2,N)-RHO*NT(DEL.(NU)T)TN              00002217
C                                                                        00002227
C     COMPUTE UNBALANCED TEMPERATURE VECTOR                              00002237
66    IF (N1.NE.1) GO TO 76                                              00002247
      Q = HEAT*(SIG(1)*DV(1,1)+SIG(2)*DV(2,2)+SIG(3)*(DV(1,2)+DV(2,1))) 00002257
      IF (ITYPE.EQ.3)  Q = Q + HEAT*(DV(1,2)+DV(2,1))*(SIG(4)-SIG(3))   00002262
      DO 78 J=1,2                                                        00002267
      DLTEE(J) = 0.D0                                                    00002277
      DO 78 I=1,NEL                                                      00002287
78    DLTEE(J) =DELTEE(J) + SHP(J,I)*UL(3,I)                             00002297
76    DO 77 I=1,NEL                                                      00002307
      II = (I-1)*NDF+1                                                   00002317
C                                                                        00002318
C     CONVECTION TERM SAME FOR 2D AND AXISYMMETRIC FLOW                  00002319
C                                                                        00002320
      P(II) = P(II) - RHO*SHP(3,I)*(V(1)*DV(1,1)+V(2)*DV(1,2))*WGT      00002321
      P(II+1) = P(II+1)-RHO*SHP(3,I)*(V(1)*DV(2,1)+V(2)*DV(2,2))*WGT    00002322
      IF (ITYPE.EQ.3) GO TO 79                                           00002324
      P(II) = P(II)-(SHP(1,I)*(SIG(1)+SIG(7))+SHP(2,I)*SIG(3))*WGT      00002327
      P(II+1) = P(II+1)-(SHP(2,I)*(SIG(2)+SIG(7))+SHP(1,I)*SIG(3))*WGT  00002337
      GO TO 80                                                           00002342
79    P(II) = P(II)-(SHP(1,I)*(SIG(1)+SIG(7))+SHP(3,I)*SIG(3)           00002343
     1 +SHP(2,I)*SIG(4))*WGT                                             00002344
      P(II+1) = P(II+1)-(SHP(2,I)*(SIG(2)+SIG(7))+SHP(1,I)*SIG(4))*WGT  00002345
80    IF (K2.EQ.3.OR.K2.EQ.4) P(II) = P(II)-(SHP(3,I)*(ELAS1(LL,1,N)+  00002347
     1 ELAS3(LL,2,N)))*WGT                                               00002357
      IF (K2.EQ.3.OR.K2.EQ.4) P(II+1) = P(II+1) - (SHP(3,I)*(           00002367
     1 ELAS2(LL,2,N)+ELAS3(LL,1,N)))*WGT                                 00002377
      IF (N1.EQ.1) A1 = Q*SHP(3,I)                                       00002387
      IF (N1.EQ.1) A2 = XK*DOT(SHP(1,I),DLTEE,NDM)                       00002397
77    IF (N1.EQ.1) P(II+NDM) = P(II+NDM) + A1*WGT - A2*WGT              00002407
65    CONTINUE                                                           00002417
33    CONTINUE                                                           00002427
5     RETURN                                                             00002442
      END                                                                00002447
      SUBROUTINE ELMT06( D , UL , XL , IX , TL , S , P ,NDF,NDM,NST,ISW)00000010
C                                                                        00000020
C**************************           *****************************00000030
C**************************    ELMT06 *****************************00000040
C**************************           *****************************00000050
C                                                                        00000055
C     AN ELEMENT FOR INTERPOLATING DISPLACEMENT, TEMPERATURE, AND STRESS00000060
C     FOR VISCOELASTICITY:  2D FLOW, OLDROYD DERIVATIVE                  00000070
C                                                                        00000080
      IMPLICIT REAL*8(A-H,O-Z)                                           00000090
      REAL*8 XMT(4,6)/8*1.D0,2*0.D0,2*1.D0,12*0.D0/,                     00000100
     1 XM(6,4)/2*1.D0,4*0.D0,2*1.D0,4*0.D0,3*1.D0,3*0.D0,3*1.D0,3*0.D0/ 00000110
      INTEGER LB(4)/3,3,4,6/                                             00000120
      COMMON /CDATA/O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR            00000130
      COMMON /ELDATA/DM,N,MA,MCT,IEL,NEL                                 00000140
      DIMENSION D(30),UL(NDF,1),XL(NDM,1),IX(1),TL(1),                   00000150
     1   S(NST,1),P(1),SHP(3,9),SG(9),TG(9),WG(9),                       00000160
     2   SIG(7),EPS(6),BSIG(3),XX(3),B(18),DB(6,3),BTDB(3,3),            00000170
     3   BU(6),XMTB(3),XMTBT(3),PEN(3,3),DU(3),DLTEE(3),                 00000180
     4   V(2),DV(2,2),XN(3,3),ADVEC(2,2),CADVEC(2,2),DDV(3,2),C(3,2),    00000190
     5   XNTN(3,3),BT(2,3),ADSIG(3,2),CN(3,2),XNTDB(3,2),XNTCN(3,2),     00000200
     6   XNTBT(2,3), CADSIG(3,2)                                          00000205
C                                                                        00000210
      IF (ISW.EQ.1) GO TO 1                                              00000220
```

```
      ITYPE = D(30)
      L = D(28)
      RHO = D(27)
      XLAM = D(26)
      XMU = D(26)
      XK = D(24)
      C9= D(23)
      N1 = D(20)
      HEAT = D(19)
      LLB = LB(ITYPE)
      K2 = D(18)
      N3 = D(17)
      G = D(16)
      P4 =  D(15)
C
C     BRANCH TO CORRECT ARRAY PROCESSOR
C
      GO TO (1,2,3,3,5,3),ISW
C
C     ISW = 1: READ MATERIAL PROPERTIES, DEVELOP
C     DIAGONAL-STORAGE D MATRIX
C
    1 CALL DFMTRX(D)
      LINT = 0
C
      RETURN
C
    2 RETURN
C
C     ISW = 3:  FORM ELEMENT STIFFNES MATRIX
C
    3 CONTINUE
C
C     LOOP OVER GAUSS INTEGRATION POINTS
C     COMPUTE UNSYMMETRIC STIFFNESS MATRIX
C
      IF (L**NDM.NE.LINT) CALL PGAUSS (L,LINT,SG,TG,WG)
      DO 33 LL = 1,LINT
C
      CALL SHAPE (SG(LL),TG(LL),XL,SHP,XSJ,NDM,NEL,IX,.FALSE.)
      WGT = XSJ*WG(LL)
C
C     COMPUTE COORDINTAES,VELOCITIES, STRESSES, AND GRADIENTS
C
      DO 32 I=1,NDM
         XX(I)=0.D0
         V(I)=0.D0
      DO 31 K=1,NEL
         XX(I) = XX(I) + SHP(3,K)*XL(I,K)
         V(I) = V(I) + SHP(3,K)*UL(I,K)
 31   CONTINUE
      DO 32 J=1,NDM
      DV(I,J)=0.D0
      DO 32 K=I,NEL
 32   DV(I,J) = DV(I,J) + SHP(J,K)*UL(I,K)
C
C     COMPUTE NONLINEAR VISCOSITY CORRECTION
C
      XNLNR = 1.D0
      IF (P4.EQ.1.) GO TO 325
```

```
      A1 = 2.D0*(DV(1,1)**2+DV(2,2)**2)+(DV(1,2)+DV(2,1))**2
      XNLNR = XNLNR/(1.D0+A1**((1.D0-P4)/2.D0))
325   VISLAM = 0.D0
      IF (G.EQ.0.D0) GO TO 9
      VISLAM = XNLNR*XMU/G + VISLAM
9     DO 320 I=1,3
      SIG(I) = 0.D0
      DO 320 J=1,2
      DDV(I,J) = 0.D0
      DO 320 K=1,NEL
      SIG(I) = SIG(I) + SHP(3,K)*UL(NDF-3+I,K)
320   DDV(I,J) = DDV(I,J) + SHP(J,K)*UL(NDF-3+I,K)
      IF (ISW.EQ.4.OR.ISW.EQ.6) GO TO 47
C
C     LOOP OVER COLUMNS FORMIN MT*B,(DEL.(NU)T)T*N, BT,
C     DEL(N*SIGMA)*N, N, DB, AND CN
C
      DO 46 J=1,NEL
      CALL BMATRX(B,J,ITYPE,SHP,RR)
      CALL CMATRX(C,J,SIG,SHP)
      CALL VMULFF(XMT(ITYPE,1),B,1,LLB,NDF-3,4,LLB,XMTB,1,IER)
      DO 37 IDEX=1,2
      DO 37 JDEX=1,2
37    ADVEC(IDEX,JDEX) = DV(IDEX,JDEX)*SHP(3,J)
      DO 41 IDEX=1,3
      DO 41 JDEX=1,3
      IF (IDEX.EQ.JDEX) XN(IDEX,JDEX) = SHP(3,J)
41    IF (IDEX.NE.JDEX) XN(IDEX,JDEX) = 0.D0
      BT(1,1) = SHP(1,J)
      BT(2,1) = 0.D0
      BT(1,2) = 0.D0
      BT(2,2) = SHP(2,J)
      BT(1,3) = SHP(2,J)
      BT(2,3) = SHP(1,J)
      DO 39 IDEX=1,3
      DO 39 JDEX=1,2
      ADSIG(IDEX,JDEX) = SHP(3,J)*DDV(IDEX,JDEX)
39    CN(IDEX,JDEX) = SHP(3,J)*C(IDEX,JDEX)
      CALL VMULDF(D,LLB,B,NDM,LLB,DB,6)
      JJ = (J-1)*NDF +1
C
C     LOOP OVER ROWS, FORMING (MTB)T*MTB, NT(DEL.(NU)T)T*N,NT*BT,
C     NT(DEL(N*SIGMA)*N, NT*N, NT*DB, AND NT*CN
C
      DO 45 I=1,NEL
C
      CALL BMATRX(B,I,ITYPE,SHP,RR)
      CALL VMULFF(XMT(ITYPE,1),B,1,LLB,NDF-3,4,LLB,XMTBT,1,IER)
      CALL VMULFM(XMTBT,XMTB,1,NDM,NDM,1,1,PEN,3,IER)
      DO 38 IDEX=1,2
      DO 38 JDEX=1,2
38    CADVEC(IDEX,JDEX) = ADVEC(IDEX,JDEX)*SHP(3,I)
      DO 40 IDEX=1,2
      DO 40 JDEX=1,3
      XNTCN(JDEX,IDEX) = CN(JDEX,IDEX)*SHP(3,I)
      XNTDB(JDEX,IDEX) = DB(JDEX,IDEX)*SHP(3,I)
      XNTBT(IDEX,JDEX) = BT(IDEX,JDEX)*SHP(3,I)
40    CADSIG(JDEX,IDEX)  = ADSIG(JDEX,IDEX)*SHP(3,I)
      DO 42 IDEX=1,3
      DO 42 JDEX=1,3
```

```
42    XNTN(IDEX,JDEX) = XN(IDEX,JDEX)*SHP(3,I)
      II = (I-1)*NDF + 1
C
C     ADD TO ELEMENT STIFFNESS MATRIX S(NST,NST)
C
      CALL MXADD(S(II,JJ),NST,PEN,3,NDM,NDM,WGT*XLAM)
      CALL MXADD(S(II,JJ),NST,CADVEC,NDM,NDM,NDM,WGT*RHO)
      CALL MXADD(S(II,JJ+NDF-2),NST,XNTBT,2,2,3,WGT)
      CALL MXADD(S(II+NDF-2,JJ),NST,CADSIG,3,3,2,WGT*VISLAM)
      CALL MXADD(S(II+NDF-2,JJ),NST,XNTCN,3,3,2,-WGT*VISLAM)
      CALL MXADD(S(II+NDF-2,JJ),NST,XNTDB,3,3,2,-WGT*XNLNR)
      CALL MXADD(S(II+NDF-2,JJ+NDF-2),NST,XNTN,3,3,3,WGT)
C
C     ADD THERMAL STIFFNESS
C
      IF(N1.EQ.1) A2 = XK*DOT(SHP(1,I),SHP(1,J),NDM)
      IF(N1.EQ.1) S(II+NDM,JJ+NDM) = S(II+NDM,JJ+NDM)+A2*WGT
45    CONTINUE
46    CONTINUE
47    CONTINUE
      IF (ISW.EQ.3) GO TO 65
C
C     PRINT STRESSES IF ISW=4,OTHERWISE BRANCH TO COMPUTE
C     UNBALANCED FORCE VECTOR
C
      IF (ISW.EQ.6) GO TO 66
      XMAX = DMAX1(DABS(XL(1,4)-XL(1,1)),DABS(XL(1,3)-XL(1,2)),
     1 DABS(XL(2,4)-XL(2,1)),DABS(XL(2,3)-XL(2,2)))
      SIG(5) = (RHO/(XMU*XNLNR))*DSQRT(V(1)**2+V(2)**2)*XMAX
      SIG(6) = VISLAM*DSQRT(V(1)**2+V(2)**2)/XMAX
      CALL FPSIG(XX,0.D0,0.D0,0.D0,SIG,ITYPE,NDF)
      GO TO 65
C
C     LOOP OVER NODES TO COMPUTE UNBALANCED FORCE VECTORS
C     P1 = P1 - NTBT*SIGMA - RHO*NT(DEL.(NU)T)T*N
C     P2 = P2 - NT(DEL(SIGMA))*V - NT*SIGMA + NT*D*L*V + NT*NT*V,I*SIGMA
C
C     COMPUTE UNBALANCED TEMPERATURE VECTOR
66    IF (N1.NE.1) GO TO 76
      Q = HEAT*(SIG(1)*DV(1,1)+SIG(2)*DV(2,2))
      DO 78 J=1,2
      DLTEE(J) = 0.D0
      DO 78 I=1,NEL
78    DLTEE(J) = DLTEE(J) + SHP(J,I)*UL(3,I)
76    DO 77 I=1,NEL
      II = (I-1)*NDF + 1
      P(II) = P(II)-(RHO*(V(1)*DV(1,1)+V(2)*DV(1,2))+(DDV(1,1)
     1        +DDV(3,2)))*SHP(3,I)*WGT - SHP(1,I)*SIG(7)*WGT
      P(II+1) = P(II+1)-(RHO*(V(1)*DV(2,1)+V(2)*DV(2,2))+(DDV(2,2)
     1        +DDV(3,1)))*SHP(3,I)*WGT - SHP(2,I)*SIG(7)*WGT
      IF (N1.EQ.1) A1 = Q*SHP(3,I)
      IF (N1.EQ.1) A2 = XK*DOT(SHP(1,I),DLTEE,NDM)
      IF (N1.EQ.1) P(II+NDM) = P(II+NDM) + A1*WGT - A2*WGT
      P(II+NDF-2) = P(II+NDF-2)-(VISLAM*(DDV(1,1)*V(1)+DDV(1,2)*V(2))
     1              + SIG(1) - 2.D0*XMU*XNLNR*DV(1,1) - 2.D0*VISLAM*
     2              (SIG(1)*DV(1,1)+SIG(3)*DV(1,2)))*SHP(3,I)*WGT
      P(II+NDF-1) = P(II+NDF-1)-(VISLAM*(DDV(2,1)*V(1)+DDV(2,2)*V(2))
     1              + SIG(2) - 2.D0*XMU*XNLNR*DV(2,2) -  .D0*VISLAM*
     2              (SIG(2)*DV(2,2)+SIG(3)*DV(2,1)))*SHP(3,I)*WGT
      P(II+NDF) = P(II+NDF) - (VISLAM*(DDV(3,1)*V(1)+DDV(3,2)*V(2))
```

```
     1          +SIG(3) - XMU*XNLNR*(DV(1,2)+DV(2,1)) -VISLAM*
     2          (SIG(2)*DV(1,2)+SIG(3)*DV(1,1)+SIG(1)*DV(2,1)
     3          +SIG(3)*DV(2,2)))*SHP(3,I)*WGT
 77   CONTINUE
 65   CONTINUE
 33   CONTINUE
 5    RETURN
      END
      SUBROUTINE ESHAP(SS,TT,X,ESHP,NDM,NEL,IX)
C
C*********************                ****************************
C*********************      ESHAP     ****************************
C*********************                ****************************
C
      IMPLICIT REAL*8(A-H,O-Z)
C     SHAPE FUNCION ROUTINE FOR 9 NODE QUADRILATERALS FOR SECOND DER.
C
      DIMENSION ESHP(3,1),X(NDM,1),SHP(3,9),IX(1),BIG(3,3),XS(2,2),
     1 EBIG(3,3),EXS(3,2),SX(2,2),TEMP(3)
      DATA S/0.5D0/,T/1.D0/,R/2.D0/
C
C     FORM 9-NODE QUADRILATERAL SHAPE FUNCTIONS FOR SECOND DERIVATIVE
      ESHP(1,1) = S*(TT**2-TT)
      ESHP(2,1) = S*(SS**2-SS)
      ESHP(3,1) = S**2*(R*SS-T)*(R*TT-T)
      ESHP(1,2) = ESHP(1,1)
      ESHP(2,2) = S*(SS**2+SS)
      ESHP(3,2) = S**2*(R*SS+T)*(R*TT-T)
      ESHP(1,3) = S*(TT**2+TT)
      ESHP(2,3) = ESHP(2,2)
      ESHP(3,3) = S**2*(R*SS+T)*(R*TT+T)
      ESHP(1,4) = ESHP(1,3)
      ESHP(2,4) = ESHP(2,1)
      ESHP(3,4) = S**2*(R*SS-T)*(R*TT+T)
      ESHP(1,5) = -R*ESHP(1,2)
      ESHP(2,5) = T-SS**2
      ESHP(3,5) = SS*(T-R*TT)
      ESHP(1,6) = T-TT**2
      ESHP(2,6) = -R*ESHP(2,2)
      ESHP(3,6) = -TT*(R*SS+T)
      ESHP(1,7) = -R*ESHP(1,4)
      ESHP(2,7) = ESHP(2,5)
      ESHP(3,7) = -SS*(R*TT+T)
      ESHP(1,8) = ESHP(1,6)
      ESHP(2,8) = -R*ESHP(2,1)
      ESHP(3,8) = TT*(T-R*SS)
      ESHP(1,9) = -R*ESHP(1,6)
      ESHP(2,9) = -R*ESHP(2,5)
      ESHP(3,9) = R**2*SS*TT
C
C     CONSTRUCT BIG MATRIX AND ITS INVERSE
C
      CALL SHAPE(SS,TT,X,SHP,XSJ,NDM,NEL,IX,.TRUE.)
      DO 130 I=1,NDM
      DO 130 J=1,2
      XS(I,J) = 0.D0
      DO 130 K=1,NEL
 130  XS(I,J) = XS(I,J) + X(I,K)*SHP(J,K)
      BIG(1,1) = XS(1,1)**2
      BIG(2,1) = XS(2,1)**2
```

```
      BIG(3,1) = XS(1,1)*XS(2,1)
      BIG(1,2) = XS(1,2)**2
      BIG(2,2) = XS(2,2)**2
      BIG(3,2) = XS(1,2)*XS(2,2)
      BIG(1,3) = 2.D0*XS(1,1)*XS(1,2)
      BIG(2,3) = 2.D0*XS(2,1)*XS(2,2)
      BIG(3,3) = XS(1,1)*XS(2,2) + XS(1,2)*XS(2,1)
C     CALCULATE DETERMINANT OF BIG
      DET = BIG(1,1)*(BIG(2,2)*BIG(3,3)-BIG(3,2)*BIG(2,3))-BIG(2,1)*
     1 (BIG(1,2)*BIG(3,3)-BIG(1,3)*BIG(3,2))+BIG(3,1)*(BIG(1,2)*BIG(2,3)
     2 -BIG(2,2)*BIG(1,3))
C
C     FORM INVERSE
      EBIG(1,1) = (BIG(2,2)*BIG(3,3)-BIG(3,2)*BIG(2,3))/DET
      EBIG(2,1) = -(BIG(1,2)*BIG(3,3)-BIG(3,2)*BIG(1,3))/DET
      EBIG(3,1) = (BIG(1,2)*BIG(2,3)-BIG(2,2)*BIG(1,3))/DET
      EBIG(1,2) = -(BIG(2,1)*BIG(3,3)-BIG(3,1)*BIG(2,3))/DET
      EBIG(2,2) = (BIG(1,1)*BIG(3,3)-BIG(3,1)*BIG(2,3))/DET
      EBIG(3,2) = -(BIG(1,1)*BIG(2,3)-BIG(2,1)*BIG(1,3))/DET
      EBIG(1,3) = (BIG(2,1)*BIG(3,2)-BIG(3,1)*BIG(2,2))/DET
      EBIG(2,3) = -(BIG(1,1)*BIG(3,2)-BIG(3,1)*BIG(1,2))/DET
      EBIG(3,3) = (BIG(1,1)*BIG(2,2)-BIG(2,1)*BIG(1,2))/DET
C
C     FORM SECOND DERIVATIVE MATRIX
C
      DO 131 I=1,2
      DO 131 J=1,3
      EXS(J,I) =0.D0
      DO 131 K=1,NEL
  131 EXS(J,I) = EXS(J,I) + X(I,K)*ESHP(J,K)
C     FORM JACOBIAN MATRIX INVERSE
C
      SX(1,1) = XS(2,2)/XSJ
      SX(2,2) = XS(1,1)/XSJ
      SX(1,2) = -XS(1,2)/XSJ
      SX(2,1) = -XS(2,1)/XSJ
C
C     FORM GLOBAL SECOND DERIVATIVES
C
      DO 132 I=1,NEL
      TEMP(1) = ESHP(1,I)
      TEMP(2) = ESHP(2,I)
      TEMP(3) = ESHP(3,I)
      DO 133 J=1,3
      ESHP(J,I) = 0.D0
      DO 134 K=1,3
      ESHP(J,I) = ESHP(J,I) +  EBIG(J,K)*(TEMP(K)- (EXS(K,1)*(SX(1,1)*
     1 SHP(1,I)+SX(1,2)*SHP(2,I)))-(EXS(K,2)*(SX(2,1)*SHP(1,I)+SX(2,2)*
     2 SHP(2,I))))
  134 CONTINUE
  133 CONTINUE
  132 CONTINUE
      RETURN
      END
      SUBROUTINE PFORM( UL , XL , TL , LD , P , S , IE , D , ID ,
     1    X , IX , F , T , JDIAG , B , A , C ,NDF,
     2   NDM,NEN1,NST,ISW,U,UD,AFL,BFL,CFL,DFL)
C     COMPUTE ELEMENT ARRAYS AND ASSEMBLE GLOBAL ARRAYS
C
C**********************          ********************************
```

```
C***********************      PFORM      *********************************
C***********************                 *********************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL AFL,BFL,CFL,DFL
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      COMMON /ELDATA/ DM,N,MA,MCT,IEL,NEL
      COMMON /PRLOD/ PROP
      COMMON /FVISC/ K2
      COMMON /TAYLR/ ESIG1(4,2,50),ESIG2(4,2,50),ESIG3(4,2,50),
     1 YY(4,2,50),ELAS1(4,2,50),ELAS2(4,2,50),ELAS3(4,2,50),
     2 BOSIG(4,2,50)
      DIMENSION XL(NDM,1),LD(NDF,1),P(1),S(NST,1),IE(1),D(30,1),ID(NDF,1
     1),X(NDM,1),IX(NEN1,1),F(NDF,1),JDIAG(1),B(1),A(1),C(1),UL(NDF,1)
     2   ,TL(1),T(1),U(1),UD(NDF,1)
C
      IF((K2.LE.2.OR.K2.EQ.5).OR.(ISW.LE.4).OR.(NDF.GE.4)) GO TO 102
C
C     SET ITERATION PARAMETERS FOR FLUID VISCOELASTICITY
C
      NSTEP = 0
      TOL1 = 1.E+1
C
C     BEGIN VISCOELASTIC ITERATION:  LOOP ON ELEMENTS
C
5     IEL = 0
      DO 101 N = 1,NUMEL
C
C     CALCULATE ELAS WITHIN ELEMENTS USING CENTRAL DIFFERENCES;
C     THESE WILL BE USED FOR BOUNDARY ELEMENTS
C
C     GAUSS POINT 1
C
      AA = YY(4,2,N)-X(2,IX(1,N))
      BB = YY(2,2,N)-X(2,IX(1,N))
      CC = YY(2,1,N)-X(1,IX(1,N))
      DD = YY(4,1,N)-X(1,IX(1,N))
      ELAS1(1,1,N) = ((ESIG1(2,1,N)-BOSIG(1,1,N))*AA-(ESIG1(4,1,N)
     1 -BOSIG(1,1,N))*BB)/(CC*AA-BB*DD)
      ELAS1(1,2,N) = ((ESIG1(4,1,N)-BOSIG(1,1,N))*CC-(ESIG1(2,1,N)
     1 -BOSIG(1,1,N))*DD)/(CC*AA-BB*DD)
      ELAS2(1,1,N) = ((ESIG2(2,1,N)-BOSIG(1,2,N))*AA-(ESIG2(4,1,N)
     1 -BOSIG(1,2,N))*BB)/(CC*AA-BB*DD)
      ELAS2(1,2,N) = ((ESIG2(4,1,N)-BOSIG(1,2,N))*CC-(ESIG2(2,1,N)
     1 -BOSIG(1,2,N))*DD)/(CC*AA-BB*DD)
      ELAS3(1,1,N) = ((ESIG3(2,1,N)-BOSIG(1,3,N))*AA-(ESIG3(4,1,N)
     1 -BOSIG(1,3,N))*BB)/(CC*AA-BB*DD)
      ELAS3(1,2,N) = ((ESIG3(4,1,N)-BOSIG(1,3,N))*CC-(ESIG3(2,1,N)
     1 -BOSIG(1,3,N))*DD)/(CC*AA-BB*DD)
C
C     GAUSS POINT 4
C
      AA = X(2,IX(4,N))-YY(1,2,N)
      BB = YY(3,2,N)-X(2,IX(4,N))
      CC = YY(3,1,N)-X(1,IX(4,N))
      DD = X(1,IX(4,N))-YY(1,1,N)
      ELAS1(4,1,N) = ((ESIG1(3,1,N)-BOSIG(4,1,N))*AA-(BOSIG(4,1,N)
     1 -ESIG1(1,1,N))*BB)/(CC*AA-BB*DD)
      ELAS1(4,2,N) = ((BOSIG(4,1,N)-ESIG1(1,1,N))*CC-(ESIG1(3,1,N)
     1 -BOSIG(4,1,N))*DD)/(CC*AA-BB*DD)
```

```
      ELAS2(4,1,N) = ((ESIG2(3,1,N)-BOSIG(4,2,N))*AA-(BOSIG(4,2,N)
     1 -ESIG2(1,1,N))*BB)/(CC*AA-BB*DD)
      ELAS2(4,2,N) = ((BOSIG(4,2,N)-ESIG2(1,1,N))*CC-(ESIG2(3,1,N)
     1 -BOSIG(4,2,N))*DD)/(CC*AA-BB*DD)
      ELAS3(4,1,N) = ((ESIG3(3,1,N)-BOSIG(4,3,N))*AA-(BOSIG(4,3,N)
     1 -ESIG3(1,1,N))*BB)/(CC*AA-BB*DD)
      ELAS3(4,2,N) = ((BOSIG(4,3,N)-ESIG3(1,1,N))*CC-(ESIG3(3,1,N)
     1 -BOSIG(4,3,N))*DD)/(CC*AA-BB*DD)
C
C     GAUSS POINT 3
C
      AA = X(2,IX(3,N))-YY(2,2,N)
      BB = X(2,IX(3,N))-YY(4,2,N)
      CC = X(1,IX(3,N))-YY(4,1,N)
      DD = X(1,IX(3,N))-YY(2,1,N)
      ELAS1(3,1,N) = ((BOSIG(3,1,N)-ESIG1(4,1,N))*AA-(BOSIG(3,1,N)
     1 -ESIG1(2,1,N))*BB)/(CC*AA-BB*DD)
      ELAS1(3,2,N) = ((BOSIG(3,1,N)-ESIG1(2,1,N))*CC-(BOSIG(3,1,N)
     1 -ESIG1(4,1,N))*DD)/(CC*AA-BB*DD)
      ELAS2(3,1,N) = ((BOSIG(3,2,N)-ESIG2(4,1,N))*AA-(BOSIG(3,2,N)
     1 -ESIG2(2,1,N))*BB)/(CC*AA-BB*DD)
      ELAS2(3,2,N) = ((BOSIG(3,2,N)-ESIG2(2,1,N))*CC-(BOSIG(3,2,N)
     1 -ESIG2(4,1,N))*DD)/(CC*AA-BB*DD)
      ELAS3(3,1,N) = ((BOSIG(3,3,N)-ESIG3(4,1,N))*AA-(BOSIG(3,3,N)
     1 -ESIG3(2,1,N))*BB)/(CC*AA-BB*DD)
      ELAS3(3,2,N) = ((BOSIG(3,3,N)-ESIG3(2,1,N))*CC-(BOSIG(3,3,N)
     1 -ESIG3(4,1,N))*DD)/(CC*AA-BB*DD)
C
C     GAUSS POINT 2
C
      AA = YY(3,2,N)-X(2,IX(2,N))
      BB = X(2,IX(2,N))-YY(1,2,N)
      CC = X(1,IX(2,N))-YY(1,1,N)
      DD = YY(3,1,N)-X(1,IX(2,N))
      ELAS1(2,1,N) = ((BOSIG(2,1,N)-ESIG1(1,1,N))*AA-(ESIG1(3,1,N)
     1 -BOSIG(2,1,N))*BB)/(CC*AA-BB*DD)
      ELAS1(2,2,N) = ((ESIG1(3,1,N)-BOSIG(2,1,N))*CC-(BOSIG(2,1,N)
     1 -ESIG1(1,1,N))*DD)/(CC*AA-BB*DD)
      ELAS2(2,1,N) = ((BOSIG(2,2,N)-ESIG2(1,1,N))*AA-(ESIG2(3,1,N)
     1 -BOSIG(2,2,N))*BB)/(CC*AA-BB*DD)
      ELAS2(2,2,N) = ((ESIG2(3,1,N)-BOSIG(2,2,N))*CC-(BOSIG(2,2,N)
     1 -ESIG2(1,1,N))*DD)/(CC*AA-BB*DD)
      ELAS3(2,1,N) = ((BOSIG(2,3,N)-ESIG3(1,1,N))*AA-(ESIG3(3,1,N)
     1 -BOSIG(2,3,N))*BB)/(CC*AA-BB*DD)
      ELAS3(2,2,N) = ((ESIG3(3,1,N)-BOSIG(2,3,N))*CC-(BOSIG(2,3,N)
     1 -ESIG3(1,1,N))*DD)/(CC*AA-BB*DD)
C
C     REPLACE ELAS FOR INTERIOR ELEMENTS
C
      DO 91 IDEX = 1,NUMEL
      DO 92 JDEX = 1,NUMEL
C
C     GAUSS POINT 1
C
      IF((IX(1,N).NE.IX(4,IDEX)).OR.(IX(2,N).NE.IX(3,IDEX)))GO TO 10
      IF((IX(1,N).NE.IX(2,JDEX)).OR.(IX(4,N).NE.IX(3,JDEX)))GO TO 10
      AA = YY(4,2,N)-YY(4,2,IDEX)
      BB = YY(2,2,N)-YY(2,2,JDEX)
      CC = YY(2,1,N)-YY(2,1,JDEX)
      DD = YY(4,1,N)-YY(4,1,IDEX)
```

```
      ELAS1(1,1,N) = ((ESIG1(2,1,N)-ESIG1(2,1,JDEX))*AA-(ESIG1(4,1,N)
     1 -ESIG1(4,1,IDEX))*BB)/(CC*AA-BB*DD)
      ELAS1(1,2,N) = ((ESIG1(4,1,N)-ESIG1(4,1,IDEX))*CC-(ESIG1(2,1,N)
     1 -ESIG1(2,1,JDEX))*DD)/(CC*AA-BB*DD)
      ELAS2(1,1,N) = ((ESIG2(2,1,N)-ESIG2(2,1,JDEX))*AA-(ESIG2(4,1,N)
     1 -ESIG2(4,1,IDEX))*BB)/(CC*AA-BB*DD)
      ELAS2(1,2,N) = ((ESIG2(4,1,N)-ESIG2(4,1,IDEX))*CC-(ESIG2(2,1,N)
     1 -ESIG2(2,1,JDEX))*DD)/(CC*AA-BB*DD)
      ELAS3(1,1,N) = ((ESIG3(2,1,N)-ESIG3(2,1,JDEX))*AA-(ESIG3(4,1,N)
     1 -ESIG3(4,1,IDEX))*BB)/(CC*AA-BB*DD)
      ELAS3(1,2,N) = ((ESIG3(4,1,N)-ESIG3(4,1,IDEX))*CC-(ESIG3(2,1,N)
     1 -ESIG3(2,1,JDEX))*DD)/(CC*AA-BB*DD)
C
C     GAUSS POINT 4
C
10    IF((IX(1,N).NE.IX(2,IDEX)).OR.(IX(4,N).NE.IX(3,IDEX)))GO TO 20
      IF((IX(3,N).NE.IX(2,JDEX)).OR.(IX(4,N).NE.IX(1,JDEX)))GO TO 20
      AA = YY(1,2,JDEX)-YY(1,2,N)
      BB = YY(3,2,N)-YY(3,2,IDEX)
      CC = YY(3,1,N)-YY(3,1,IDEX)
      DD = YY(1,1,JDEX)-YY(1,1,N)
      ELAS1(4,1,N) = ((ESIG1(3,1,N)-ESIG1(3,1,IDEX))*AA-(ESIG1(1,1,JDEX)
     1 -ESIG1(1,1,N))*BB)/(CC*AA-BB*DD)
      ELAS1(4,2,N) = ((ESIG1(1,1,JDEX)-ESIG1(1,1,N))*CC-(ESIG1(3,1,N)
     1 -ESIG1(3,1,IDEX))*DD)/(CC*AA-BB*DD)
      ELAS2(4,1,N) = ((ESIG2(3,1,N)-ESIG2(3,1,IDEX))*AA-(ESIG2(1,1,JDEX)
     1 -ESIG2(1,1,N))*BB)/(CC*AA-BB*DD)
      ELAS2(4,2,N) = ((ESIG2(1,1,JDEX)-ESIG2(1,1,N))*CC-(ESIG2(3,1,N)
     1 -ESIG2(3,1,IDEX))*DD)/(CC*AA-BB*DD)
      ELAS3(4,1,N) = ((ESIG3(3,1,N)-ESIG3(3,1,IDEX))*AA-(ESIG3(1,1,JDEX)
     1 -ESIG3(1,1,N))*BB)/(CC*AA-BB*DD)
      ELAS3(4,2,N) = ((ESIG3(1,1,JDEX)-ESIG3(1,1,N))*CC-(ESIG3(3,1,N)
     1 -ESIG3(3,1,IDEX))*DD)/(CC*AA-BB*DD)
C
C     GAUSS POINT 3
C
20    IF((IX(3,N).NE.IX(2,IDEX)).OR.(IX(4,N).NE.IX(1,IDEX)))GO TO 30
      IF((IX(2,N).NE.IX(1,JDEX)).OR.(IX(2,N).NE.IX(4,JDEX)))GO TO 30
      AA = YY(2,2,IDEX)-YY(2,2,N)
      BB = YY(4,2,JDEX)-YY(4,2,N)
      CC = YY(4,1,JDEX)-YY(4,1,N)
      DD = YY(2,1,IDEX)-YY(2,1,N)
      ELAS1(3,1,N) = ((ESIG1(4,1,JDEX)-ESIG1(4,1,N))*AA-(ESIG1(2,1,IDEX)
     1 -ESIG1(2,1,N))*BB)/(CC*AA-BB*DD)
      ELAS1(3,2,N) = ((ESIG1(2,1,IDEX)-ESIG1(2,1,N))*CC-(ESIG1(4,1,JDEX)
     1 -ESIG1(4,1,N))*DD)/(CC*AA-BB*DD)
      ELAS2(3,1,N) = ((ESIG2(4,1,JDEX)-ESIG2(4,1,N))*AA-(ESIG2(2,1,IDEX)
     1 -ESIG2(2,1,N))*BB)/(CC*AA-BB*DD)
      ELAS2(3,2,N) = ((ESIG2(2,1,IDEX)-ESIG2(2,1,N))*CC-(ESIG2(4,1,JDEX)
     1 -ESIG2(4,1,N))*DD)/(CC*AA-BB*DD)
      ELAS3(3,1,N) = ((ESIG3(4,1,JDEX)-ESIG3(4,1,N))*AA-(ESIG3(2,1,IDEX)
     1 -ESIG3(2,1,N))*BB)/(CC*AA-BB*DD)
      ELAS3(3,2,N) = ((ESIG3(2,1,IDEX)-ESIG3(2,1,N))*CC-(ESIG3(4,1,JDEX)
     1 -ESIG3(4,1,N))*DD)/(CC*AA-BB*DD)
C
C     GAUSS POINT 2
C
30    IF((IX(2,N).NE.IX(1,IDEX)).OR.(IX(3,N).NE.IX(4,IDEX)))GO TO 92
      IF((IX(1,N).NE.IX(4,JDEX)).OR.(IX(2,N).NE.IX(3,JDEX)))GO TO 92
      AA = YY(3,2,N)-YY(3,2,JDEX)
```

```
      BB = YY(1,2,IDEX)-YY(1,2,N)
      CC = YY(1,1,IDEX)-YY(1,1,N)
      DD = YY(3,1,N)-YY(3,1,JDEX)
      ELAS1(2,1,N) = ((ESIG1(1,1,IDEX)-ESIG1(1,1,N))*AA-(ESIG1(3,1,N)
     1 -ESIG1(3,1,JDEX))*BB)/(CC*AA-BB*DD)
      ELAS1(2,2,N) = ((ESIG1(3,1,N)-ESIG1(3,1,JDEX))*CC-(ESIG1(1,1,IDEX)
     1 -ESIG1(1,1,N))*DD)/(CC*AA-BB*DD)
      ELAS2(2,1,N) = ((ESIG2(1,1,IDEX)-ESIG2(1,1,N))*AA-(ESIG2(3,1,N)
     1 -ESIG2(3,1,JDEX))*BB)/(CC*AA-BB*DD)
      ELAS2(2,2,N) = ((ESIG2(3,1,N)-ESIG2(3,1,JDEX))*CC-(ESIG2(1,1,IDEX)
     1 -ESIG2(1,1,N))*DD)/(CC*AA-BB*DD)
      ELAS3(2,1,N) = ((ESIG3(1,1,IDEX)-ESIG3(1,1,N))*AA-(ESIG3(3,1,N)
     1 -ESIG3(3,1,JDEX))*BB)/(CC*AA-BB*DD)
      ELAS3(2,2,N) = ((ESIG3(3,1,N)-ESIG3(3,1,JDEX))*CC-(ESIG3(1,1,IDEX)
     1 -ESIG3(1,1,N))*DD)/(CC*AA-BB*DD)
   92 CONTINUE
   91 CONTINUE
C     SET UP LOCAL ARRAYS FOR CALCULATING ESIG(LL,2,N)
      DO 58 I=1,NEN
      II = IX(I,N)
      IF (II.NE.0) GO TO 55
      TL(I) = 0.
      DO 53 J=1,NDM
   53 XL(J,I) = 0.
      DO 54 J=1,NDM
      UL(J,I) = 0.
      UL(J,I+NEN) = 0.
   54 LD(J,I) = 0
      GO TO 58
   55 IID = II*NDF-NDF
      NEL = I
      TL(I) = T(II)
      DO 56 J=1,NDM
   56 XL(J,I) = X(J,II)
      DO 57 J=1,NDF
      K = IABS(ID(J,II))
      UL(J,I) = F(J,II)*PROP
      UL(J,I+NEN) = UD(J,II)
      IF (K.GT.0) UL(J,I) = U(K)
      IF (DFL) K = IID + J
   57 LD(J,I) = K
   58 CONTINUE
C     FORM ELEMENT ARRAY
      MA = IX(NEN1,N)
      IF (IE(MA).NE.IEL) MCT = 0
      IEL = IE(MA)
      CALL ELMLIB(D(1,MA),UL,XL,IX(1,N),TL,S,P,NDF,NDM,NST,7)
  101 CONTINUE
      YMAX =DMAX1(DABS(ESIG1(1,2,1)-ESIG1(1,1,1)),DABS(ESIG2(1,2,1)
     1 -ESIG2(1,1,1)),DABS(ESIG3(1,2,1)-ESIG3(1,1,1)))
      DO 93 I=1,NUMEL
      DO 93 J=1,4
      XMAX =DMAX1(DABS(ESIG1(J,2,I)-ESIG1(J,1,I)),DABS(ESIG2(J,2,I)
     1 -ESIG2(J,1,I)),DABS(ESIG3(J,2,I)-ESIG3(J,1,I)))
   93 IF (XMAX.GT.YMAX) YMAX=XMAX
      IF (YMAX.LE.TOL1) GO TO 102
      NSTEP = NSTEP + 1
      DO 90 K=1,NUMEL
      DO 90 J=1,4
      ESIG1(J,1,K) = ESIG1(J,2,K)
```

```
      ESIG2(J,1,K) = ESIG2(J,2,K)
90    ESIG3(J,1,K) = ESIG3(J,2,K)
      IF (NSTEP.GE.10) GO TO 102
      IF (NSTEP.EQ.1.OR.NSTEP.EQ.3.OR.NSTEP.EQ.5.
     1 OR.NSTEP.EQ.9) GO TO 94
      GO TO 5
94    WRITE (6,1000) O,HEAD,TIME,NSTEP
      WRITE (6,1010)
      DO 95 I=1,NUMEL
95    WRITE (6,1020) I,((YY(J,1,I),YY(J,2,I),ESIG1(J,2,I)
     1 ,ESIG2(J,2,I),ESIG3(J,2,I)),J=1,4)
      GO TO 5
C     LOOP ON ELEMENTS:  ELASTIC ITERATION COMPLETE
102   CONTINUE
      IEL = 0
      DO 110 N = 1,NUMEL
C     SET UP LOCAL ARRAYS
      DO 108 I = 1,NEN
      II = IX(I,N)
      IF (II.NE.0) GO TO 105
      TL(I) = 0.
      DO 103 J=1,NDM
103   XL(J,I) = 0.
      DO 104 J = 1,NDF
      UL(J,I) = 0.
      UL(J,I+NEN) = 0.
104   LD(J,I) = 0
      GO TO 108
105   IID = II*NDF - NDF
      NEL = I
      TL(I) = T(II)
      DO 106 J=1,NDM
106   XL(J,I) = X(J,II)
      DO 107 J=1,NDF
      K = IABS(ID(J,II))
      UL(J,I) = F(J,II)*PROP
      UL(J,I+NEN) = UD(J,II)
      IF (K.GT.0) UL(J,I) = U(K)
      IF (DFL) K = IID + J
107   LD(J,I) = K
108   CONTINUE
C     FORM ELEMENT ARRAY
      MA = IX(NEN1,N)
      IF(IE(MA).NE.IEL) MCT = 0
      IEL = IE(MA)
      CALL ELMLIB(D(1,MA),UL,XL,IX(1,N),TL,S,P,NDF,NDM,NST,ISW)
C     ADD TO TOTAL ARRAY
      IF(AFL.OR.BFL.OR.CFL) CALL ADDSTF(A,B,C,S,P,JDIAG,LD,NST,NEL*NDF,
     1   AFL,BFL,CFL)
110   CONTINUE
1000  FORMAT(A1,20A4,//5X,'ELASTIC FLUID STRESSES AT GAUSS POINTS',
     1 5X,'TIME',G13.5,//1X,'VISCOELASTIC ITERATION NUMBER:',I4//)
1010  FORMAT(1X,'ELMT',11X,'1-COORD',6X,'2-COORD',20X,
     1 'ETAU-XX',7X,'ETAU-YY',7X,'ETAU-XY'//)
1020  FORMAT(I5,/4(10X,2G13.4,13X,3G13.4/)//)
      RETURN
      END
      SUBROUTINE CMATRX(C,J,SIG,SHP)
C
C**********************           ****************************
```

```
C***********************     CMATRX     *****************************
C***********************                *****************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION C(1),SIG(7),SHP(3,9)
C
      CALL PZERO(C,6)
C
C     ONLY 2D FLOW TREATED HERE
C
      C(1) = 2.D0*(SIG(1)*SHP(1,J)+SIG(3)*SHP(2,J))
      C(2) = 0.D0
      C(3) = SIG(2)*SHP(2,J)+SIG(3)*SHP(1,J)
      C(4) = 0.D0
      C(5) = 2.D0*(SIG(2)*SHP(2,J)+SIG(3)*SHP(1,J))
      C(6) = SIG(1)*SHP(1,J+SIG(3)*SHP(2,J))
      RETURN
      END
      SUBROUTINE FPSIG (XX,ESIG1,ESIG2,ESIG3,SIG,ITYPE,NDF)
C
C************************     FPSIG     ***************************
C************************               ***************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XX(1),SIG(1)
      COMMON /CDATA/ O,HEAD(20),NUMNP,NUMEL,NUMMAT,NEN,NEQ,IPR
      COMMON /ELDATA/ DM,N,MA,MOT,IEL,NEL
      COMMON /TDATA/ TIME,DT,C1,C2,C3,C4,C5
      COMMON /FVISC/ K2
C
      GO TO (51,52,53,54), ITYPE
C
C     PLANE FLOW
C
51    MOT=MOT-1
      IF (K2.LE.2) GO TO 509
      A = SIG(1) + ESIG1
      B = SIG(2) + ESIG2
      C = SIG(3) + ESIG3
509   IF (MOT.GT.0) GO TO 510
      IF (NDF.LT.4) WRITE (6,5000) O,HEAD,TIME
5000  FORMAT (A1,20A4,//5X,'FLUID VISCOUS STRESSES AT GAUSS POINTS:',
     1   5X,'TIME',G13.5,
     2   //1X,'ELMT MATL',6X,'1-COORD',6X,'2-COORD',5X,
     3   'PRESSURE',7X,'TAU-XX',7X,'TAU-YY',7X,'TAU-XY'/)
      IF (NDF.LT.4.AND.K2.GE.3) WRITE (6,5010)
5010  FORMAT (//5X,'TOTAL VISCOUS AND ELASTIC STRESSES ATGAUSS POINTS:
     1'/)
      IF (NDF.GE.4) WRITE (6,5020) O,HEAD,TIME
5020  FORMAT (A1,20A4,//5X,'TOTAL VISCOUS AND ELASTIC STRESSES AT
     1    GAUSS POINTS:',5X,'TIME',G13.5,
     2    //1X,'ELMT MATL',6X,'1-COORD',6X,'2-COORD',5X,
     3    'PRESSURE',7X,'TAU-XX',7X,'TAU-YY',7X,'TAU-XY'/)
      IF (NDF.GE.4) GO TO 509
      IF (K2.GE.3) MOT=19
      IF (K2.GE.3) GO TO 508
      MOT = 50
508   CONTINUE
510   IF (NDF.LT.4) WRITE (6,5001)N,MA,XX(1),XX(2),SIG(7),(SIG(I),I=1,3)
      IF (NDF.GE.4) WRITE (6,5009)N,MA,XX(1),XX(2)
```

```
 5009 FORMAT (2I5,2G13.4)
      IF (K2.GE.3) WRITE (6,5011) SIG(5),SIG(6),A,B,C
 5001 FORMAT (2I5,6G13.4)
 5011 FORMAT (1X,'RE = ',G13.4,'WS = ',G13.4,13X,3G13.4)
      RETURN
C
 52   RETURN
C
C     AXISYMMETRIC FLOW
C
 53   MOT=MOT-1
      IF (MOT.GT.0) GO TO 530
      WRITE (6,5002) O,HEAD,TIME
 5002 FORMAT (A1,20A4,//5X,'FLUID STRESSES AT GAUSS POINTS:',
     1   5X,'TIME',G13.5,
     2   //1X,'ELMT MATL',6X,'1-COORD',6X,'2-COORD',5X,
     3   'PRESSURE',7X,'TAU-RR',7X,'TAU-ZZ',7X,'TAU-TT',7X,'TAU-RZ'/)
      MOT = 50
C
 530  WRITE (6,5003) N,MA,XX(1),XX(2),SIG(7),(SIG(I),I=1,4)
 5003 FORMAT (2I5,7G13.4)
      RETURN
C
 54   RETURN
      END
```

APPENDIX 4

Input Data Set Listings

1. Run 1 - Linear Cross Channel Flow
   18-9 Node Elements
2. Run 3 - Linear Cross Channel Flow
   72-8 Node Elements
3. Run 4 - Convection (Re = 0.4) Cross Channel Flow
   18-9 Node Elements
4. Run 6 - Viscoelastic (Ws = 0.02) Cross Channel Flow
   18-9 Node Elements
5. Run 13 - Viscoelastic (Ws = 0.001) Entry Flow
   24-9 Node Elements
6. Run 20 - Linear Entry Flow Fully Developed Boundary
   Conditions 24-9 Node Elements
   (Note: Run 20 is not listed in Table 1)

Input Dataset Run No. 1

```
BRC4066  (FOREGROUND):  OUTPUT FROM TSO XPRINT

AT 13:42:06 ON 12/05/80 - BRC4066.TEST.DATA

FEAP  CROSS-CHANNEL FLOW - NEWTONIAN (TEST 7)                                 00000010
   91   18    1    2    2    9    0                                           00000020
COOR                                                                          00000030
    1    7        0D0        0D0                                              00000040
   85    0        2D0        0D0                                              00000050
    2    7        0D0.1666667D0                                               00000060
   86    0        2D0.1666667D0                                               00000070
    3    7        0D0.3333333D0                                               00000080
   87    0        2D0.3333333D0                                               00000090
    4    7        0D0       .5D0                                              00000100
   88    0        2D0       .5D0                                              00000110
    5    7        0D0.6666667D0                                               00000120
   89    0        2D0.6666667D0                                               00000130
    6    7        0D0.8333333D0                                               00000140
   90    0        2D0.8333333D0                                               00000150
    7    7        0D0        1D0                                              00000160
   91    0        2D0        1D0                                              00000170
                                                                              00000180
ELEM                                                                          00000190
    1    1    1   15   17    3    8   16   10    2    9   14                  00000200
    7    1    3   17   19    5   10   18   12    4   11   14                  00000210
   13    1    5   19   21    7   12   20   14    6   13   14                  00000220
                                                                              00000230
MATE                                                                          00000240
    1    5   NINE-NODE LAGRAGIAN PENALTY ELEMENT                              00000250
    1    0    2    1      1.0                                                 00000260
    2 .1000+009 .7900+003      .0000                                          00000270
                                                                              00000275
BOUN                                                                          00000280
    1    7   -1   -1                                                          00000290
   85    0    1    1                                                          00000300
    2    1   -1   -1                                                          00000310
    6    0    1    1                                                          00000320
   86    1   -1   -1                                                          00000330
   91    0    1    1                                                          00000340
    7    7   -1   -1                                                          00000350
   89    0    1    1                                                          00000360
                                                                              00000370
FORC                                                                          00000380
    7    7       -1D2        0D0                                              00000390
   91    0       -1D2        0D0                                              00000400
                                                                              00000480
END                                                                           00000490
MACR                                                                          00000500
UTAN                                                                          00000510
FORM                                                                          00000520
SOLV                                                                          00000530
DISP                                                                          00000540
STRE                                                                          00000550
REAC                                                                          00000560
END                                                                           00000570
STOP                                                                          00000580
```

Input Dataset Run No. 3

```
BRC4066  (FOREGROUND):  OUTPUT FROM TSO XPRINT

AT 13:42:48 ON 12/05/80 - BRC4066.TEST2.DATA

FEAP CROSS-CHANNEL FLOW--LINEAR NEWTONIAN//72 ELEMENTS                  00000010
  253   72    1    2    2    8    0                                     00000020
COOR                                                                    00000030
    1    1        0D0       0D0                                         00000040
   13    0        0D0       1D0                                         00000050
   14    1.0833333D0        0D0                                         00000060
   20    0.0833333D0        1D0                                         00000070
   21    1.1666667D0        0D0                                         00000080
   33    0.1666667D0        1D0                                         00000090
   34    1      .25D0       0D0                                         00000100
   40    0      .25D0       1D0                                         00000110
   41    1.3333333D0        0D0                                         00000120
   53    0.3333333D0        1D0                                         00000130
   54    1.4166667D0        0D0                                         00000140
   60    0.4166667D0        1D0                                         00000150
   61    1       .5D0       0D0                                         00000160
   73    0       .5D0       1D0                                         00000170
   74    1.5833333D0        0D0                                         00000180
   80    0.5833333D0        1D0                                         00000190
   81    1.6666667D0        0D0                                         00000200
   93    0.6666667D0        1D0                                         00000210
   94    1      .75D0       0D0                                         00000220
  100    0      .75D0       1D0                                         00000230
  101    1.8333333D0        0D0                                         00000240
  113    0.8333333D0        1D0                                         00000250
  114    1.9166667D0        0D0                                         00000260
  120    0.9166667D0        1D0                                         00000270
  121    1        1D0       0D0                                         00000280
  133    0        1D0       1D0                                         00000290
  134    11.0833333D0       0D0                                         00000300
  140    01.0833333D0       1D0                                         00000310
  141    11.1666667D0       0D0                                         00000320
  153    01.1666667D0       1D0                                         00000330
  154    1    1.25D0        0D0                                         00000340
  160    0    1.25D0        1D0                                         00000350
  161    11.3333333D0       0D0                                         00000360
  173    01.3333333D0       1D0                                         00000370
  174    11.4166667D0       0D0                                         00000380
  180    01.4166667D0       1D0                                         00000390
  181    1     1.5D0        0D0                                         00000400
  193    0     1.5D0        1D0                                         00000410
  194    11.5833333D0       0D0                                         00000420
  200    01.5833333D0       1D0                                         00000430
  201    11.6666667D0       0D0                                         00000440
  213    01.6666667D0       1D0                                         00000450
  214    1    1.75D0        0D0                                         00000460
  220    0    1.75D0        1D0                                         00000470
  221    11.8333333D0       0D0                                         00000480
  233    01.8333333D0       1D0                                         00000490
  234    11.9166667D0       0D0                                         00000500
  240    01.9166667D0       1D0                                         00000510
  241    1        2D0       0D0                                         00000520
  253    0        2D0       1D0                                         00000530
                                                                        00000540
ELEM                                                                    00000550
    1    1    1   21   23    3   14   22   15    2   20                 00000560
```

```
   13    1    3   23   25    5   15   24   16    4   20
   25    1    5   25   27    7   16   26   17    6   20
   37    1    7   27   29    9   17   28   18    8   20
   49    1    9   29   31   11   18   30   19   10   20
   61    1   11   31   33   13   19   32   20   12   20

MATE
    1    5    EIGHT-NODE SERENDIPITY PENALTY ELEMENT
    1    0    1    1       1.0
    2 .1000+009 .7900+003     .0000

BOUN
    1    1   -1   -1
   13    0    1    1
   14   20   -1   -1
  234    0    1    1
   21   20   -1   -1
  221    0    1    1
  241    1   -1   -1
  253    0    1    1
   20   20   -1   -1
  240    0    1    1
   33   20   -1   -1
  233    0    1    1

FORC
   20   20      -1D2       0D0
  240    0      -1D2       0D0
   13   20      -1D2       0D0
  253    0      -1D2       0D0

END
MACR
TANG
FORM
SOLV
DISP
STRE
END
STOP
```

Input Dataset Run No. 4

```
BRC4066  (FOREGROUND):  OUTPUT FROM TSO XPRINT

AT 16:22:13 ON 12/05/80 - BRC4066.TEST.DATA

FEAP  CROSS-CHANNEL FLOW - NEWTONIAN WITH CONVECTION                       00000010
   91   18    1    2    2    9    0                                        00000020
COOR                                                                       00000030
    1    7        000        000                                           00000040
   85    0        200        000                                           00000050
    2    7        000.1666666700                                           00000060
   86    0        200.1666666700                                           00000070
    3    7        000.3333333300                                           00000080
   87    0        200.3333333300                                           00000090
    4    7        000        .500                                          00000100
   88    0        200        .500                                          00000110
    5    7        000.6666666700                                           00000120
   89    0        200.6666666700                                           00000130
    6    7        000.8333333300                                           00000140
   90    0        200.8333333300                                           00000150
    7    7        000        100                                           00000160
   91    0        200        100                                           00000170
                                                                           00000180
ELEM                                                                       00000190
    1    1    1   15   17    3    8   16   10    2    9   14               00000200
    7    1    3   17   19    5   10   18   12    4   11   14               00000210
   13    1    5   19   21    7   12   20   14    6   13   14               00000220
                                                                           00000230
MATE                                                                       00000240
    1    5    NINE-NODE LAGRAGIAN PENALTY ELEMENT                          00000250
    1    0    1    1        1.0                                            00000260
    2 .1000+009 .7900+003    1.6000                                        00000270
                                                                           00000275
BOUN                                                                       00000280
    1    7   -1   -1                                                       00000290
   85    0    1    1                                                       00000300
    2    1   -1   -1                                                       00000310
    6    0    1    1                                                       00000320
   86    1   -1   -1                                                       00000330
   91    0    1    1                                                       00000340
    7    7   -1   -1                                                       00000350
   89    0    1    1                                                       00000360
                                                                           00000370
FORC                                                                       00000380
    7    7      -102        000                                            00000390
   91    0      -102        000                                            00000400
                                                                           00000480
END                                                                        00000490
MACR                                                                       00000500
DT         1.                                                              00000505
LOOP          3                                                            00000520
UTAN                                                                       00000530
FORM                                                                       00000540
SOLV                                                                       00000550
DISP          1                                                            00000560
STRE          1                                                            00000565
TIME                                                                       00000567
NEXT                                                                       00000570
DISP                                                                       00000580
STRE                                                                       00000590
REAC                                                                       00000595
```

```
      END                                                           00000600
      STOP                                                          00000610
```

```
BRC4066  (FOREGROUND):  OUTPUT FROM TSO XPRINT

AT 13:50:54 ON 12/07/80 - BRC4066.TEST.DATA

FEAP  SQUARE CAVITY- OLDROYD VISCOELASTIC (RHO=0, P4=1, WS=.01)          00000010
  91   18    1    2    2    9    0                                      00000020
COOR                                                                    00000030
   1    7        0D0        0D0                                         00000040
  85    0        2D0        0D0                                         00000050
   2    7        0D0.1666667D0                                          00000060
  86    0        2D0.1666667D0                                          00000070
   3    7        0D0.3333333D0                                          00000080
  87    0        2D0.3333333D0                                          00000090
   4    7        0D0        .5D0                                        00000100
  88    0        2D0        .5D0                                        00000110
   5    7        0D0.6666667D0                                          00000120
  89    0        2D0.6666667D0                                          00000130
   6    7        0D0.8333333D0                                          00000140
  90    0        2D0.8333333D0                                          00000150
   7    7        0D0        1D0                                         00000160
  91    0        2D0        1D0                                         00000170
                                                                        00000180
ELEM                                                                    00000190
   1    1    1   15   17    3    8   16   10    2    9   14             00000200
   7    1    3   17   19    5   10   18   12    4   11   14             00000210
  13    1    5   19   21    7   12   20   14    6   13   14             00000220
                                                                        00000230
MATE                                                                    00000240
   1    5    NINE-NODE LAGRAGIAN PENALTY ELEMENT                        00000250
   1    0    3    1        1.0                                          00000260
   2 .1000+009 .7900+003              .3950+007                         00000270
                                                                        00000275
BOUN                                                                    00000280
   1    7   -1   -1                                                     00000290
  85    0    1    1                                                     00000300
   2    1   -1   -1                                                     00000310
   6    0    1    1                                                     00000320
  86    1   -1   -1                                                     00000330
  91    0    1    1                                                     00000340
   7    7   -1   -1                                                     00000350
  89    0    1    1                                                     00000360
                                                                        00000370
FORC                                                                    00000380
   7    7        -1D2        0D0                                        00000390
  91    0        -1D2        0D0                                        00000400
                                                                        00000485
END                                                                     00000490
MACR                                                                    00000500
DT          1.                                                          00000510
LOOP         20                                                         00000520
UTAN                                                                    00000530
FORM                                                                    00000540
SOLV                                                                    00000550
DISP          5                                                         00000558
STRE          5                                                         00000566
TIME                                                                    00000575
NEXT                                                                    00000585
DISP                                                                    00000588
STRE                                                                    00000591
REAC                                                                    00000595
```

```
      END                                                              00000600
      STOP                                                             00000610
```

Input Dataset Run No. 13

```
BRC4066  (FOREGROUND):  OUTPUT FROM TSO XPRINT

AT 15:38:57 ON 12/07/80 - BRC4066.TEST3.DATA

FEAP ENTRY FLOW - OLDROYD VISCOELASTIC (RHO=0, P4=1, WS=0.0005)        00000010
  121   24    1    2    2    9    0                                     00000020
COOR                                                                    00000030
    1    1       0D0       0D0                                          00000040
    9    0       0D0       1D0                                          00000050
   10    1    .125D0       0D0                                          00000060
   18    0    .125D0       1D0                                          00000070
   19    1     .25D0       0D0                                          00000080
   27    0     .25D0       1D0                                          00000090
   28    1    .375D0       0D0                                          00000100
   36    0    .375D0       1D0                                          00000110
   37    1      .5D0       0D0                                          00000120
   45    0      .5D0       1D0                                          00000130
   46    1    .625D0       0D0                                          00000140
   54    0    .625D0       1D0                                          00000150
   55    1     .75D0       0D0                                          00000160
   63    0     .75D0       1D0                                          00000170
   64    1    .875D0       0D0                                          00000180
   72    0    .875D0       1D0                                          00000190
   73    1       1D0       0D0                                          00000200
   81    0       1D0       1D0                                          00000210
   82    5   1.125D0     .25D0                                          00000220
  117    0       2D0     .25D0                                          00000230
   83    5   1.125D0    .375D0                                          00000240
  118    0       2D0    .375D0                                          00000250
   84    5   1.125D0      .5D0                                          00000260
  119    0       2D0      .5D0                                          00000270
   85    5   1.125D0    .625D0                                          00000280
  120    0       2D0    .625D0                                          00000290
   86    5   1.125D0     .75D0                                          00000300
  121    0       2D0     .75D0                                          00000310
                                                                        00000320
ELEM                                                                    00000330
    1    1    1   19   21    3   10   20   12    2   11   18            00000340
    5    1    3   21   23    5   12   22   14    4   13   18            00000350
    9    1    5   23   25    7   14   24   16    6   15   18            00000360
   13    1    7   25   27    9   16   26   18    8   17   18            00000370
   17    1   75   87   89   77   82   88   84   76   83    0            00000380
   18    1   87   97   99   89   92   98   94   88   93   10            00000390
   21    1   77   89   91   79   84   90   86   78   85    0            00000400
   22    1   89   99  101   91   94  100   96   90   95   10            00000410
                                                                        00000420
MATE                                                                    00000430
    1    5   NINE-NODE LAGRANGE PENALTY ELEMENT                         00000440
    1    0    3    1       1.0                                          00000450
    2 .1000+009 .7900+003     .0000 .7950+008                           00000460
                                                                        00000470
BOUN                                                                    00000480
    1    9   -1   -1                                                    00000490
   73    0    1    1                                                    00000500
    2    1   -1   -1                                                    00000510
    8    0    1    1                                                    00000520
    9    9   -1   -1                                                    00000530
   81    0    1    1                                                    00000540
   74    0    1    1                                                    00000550
   75    0    1    1                                                    00000560
```

```
   79    0    1    1
   80    0    1    1
   82    5   -1   -1
  117    0    1    1
   86    5   -1   -1
  121    0    1    1
  118    0    0    1
  119    0    0    1
  120    0    0    1

FORC
    2    1        102        000
    8    0        102        000

END
MACR
DT           1.
LOOP          30
UTAN
FORM
SOLV
DISP           5
STRE           5
TIME
NEXT
DISP
STRE
REAC
END
STOP
```

BRC4066 (FOREGROUND): OUTPUT FROM TSO XFRINT

AT 18:59:50 ON 12/18/80 - BRC4066.TEST3.DATA

Input Dataset Run No. 20

```
FEAP ENTRY FLOW - OLDROYD VIS"OELASTIC (RHO=0, P4=1, WS=0.0005)          00000010
  121   24    1    2    2    9    0                                      00000020
COOR                                                                     00000030
    1    1        0D0        0D0                                         00000040
    9    0        0D0        1D0                                         00000050
   10    1     .125D0        0D0                                         00000060
   18    0     .125D0        1D0                                         00000070
   19    1      .25D0        0D0                                         00000080
   27    0      .25D0        1D0                                         00000090
   28    1     .375D0        0D0                                         00000100
   36    0     .375D0        1D0                                         00000110
   37    1       .5D0        0D0                                         00000120
   45    0       .5D0        1D0                                         00000130
   46    1     .625D0        0D0                                         00000140
   54    0     .625D0        1D0                                         00000150
   55    1      .75D0        0D0                                         00000160
   63    0      .75D0        1D0                                         00000170
   64    1     .875D0        0D0                                         00000180
   72    0     .875D0        1D0                                         00000190
   73    1        1D0        0D0                                         00000200
   81    0        1D0        1D0                                         00000210
   82    5    1.125D0      .25D0                                         00000220
  117    0        2D0      .25D0                                         00000230
   83    5    1.125D0     .375D0                                         00000240
  118    0        2D0     .375D0                                         00000250
   84    5    1.125D0       .5D0                                         00000260
  119    0        2D0       .5D0                                         00000270
   85    5    1.125D0     .625D0                                         00000280
  120    0        2D0     .625D0                                         00000290
   86    5    1.125D0      .75D0                                         00000300
  121    0        2D0      .75D0                                         00000310
                                                                         00000320
ELEM                                                                     00000330
    1    1    1   19   21    3   10   20   12    2   11   18            00000340
    5    1    3   21   23    5   12   22   14    4   13   18            00000350
    9    1    5   23   25    7   14   24   16    6   15   18            00000360
   13    1    7   25   27    9   16   26   18    8   17   18            00000370
   17    1   75   87   89   77   82   88   84   76   83    0            00000380
   18    1   87   97   99   89   92   98   94   88   93   10            00000390
   21    1   77   89   91   79   84   90   86   78   85    0            00000400
   22    1   89   99  101   91   94  100   96   90   95   10            00000410
                                                                         00000420
MATE                                                                     00000430
    1    5    NINE-NODE LAGRANGE PENALTY ELEMENT                         00000440
    1    0    1    1      1.0                                            00000450
    2 .1000+009 .7900+003      .0000 .7950+008                           00000460
                                                                         00000470
BOUN                                                                     00000480
    1    9   -1   -1                                                     00000490
   73    0    1    1                                                     00000500
   75    0    1    1                                                     00000503
   79    0    1    1                                                     00000506
    2    1    0   -1                                                     00000510
    8    0    0    1                                                     00000520
    9    9   -1   -1                                                     00000530
   81    0    1    1                                                     00000540
```

```
   74     0    1    1                                                00000550
   80     0    1    1                                                00000580
   82     5   -1   -1                                                00000590
  117     0    1    1                                                00000600
   86     5   -1   -1                                                00000610
  121     0    1    1                                                00000620
  118     0    0    1                                                00000630
  119     0    0    1                                                00000640
  120     0    0    1                                                00000650
                                                                     00000660
FORC                                                                 00000670
     2    0      4.2D2         0D0                                   00000680
     3    0      2.1D2                                               00000681
     4    0      4.2D2                                               00000682
     5    0      2.1D2                                               00000683
     6    0      4.2D2                                               00000684
     7    0      2.1D2                                               00000685
     8    0      4.2D2         0D0                                   00000690
                                                                     00000700
END                                                                  00000710
MACR                                                                 00000720
UTAN                                                                 00000750
FORM                                                                 00000760
SOLV                                                                 00000770
DISP                                                                 00000780
STRE                                                                 00000790
REAC                                                                 00000840
END                                                                  00000850
STOP                                                                 00000860
```

APPENDIX 5

Brief Review of Gyroscope Theory

This Appendix is presented for the benefit of the materials engineer who may not be familiar with the theory of gyroscopic behavior. The discussion is taken entirely from Wrigley et. al. [36]. Figure 18 shows a cutaway of the single degree of freedom gyroscope used in this study. The normal assumptions for the description of a gyro element performance are:

1. The rotor spins about an axis of symmetry.
2. The rotor spins at constant speed.
3. Spin angular momentum is much greater than non-spin angular momentum.
4. Center of mass of the rotor and gyro element coincide,

and 5. The rotor bearing structure is rigid.

For a platform stabilized single degree of freedom gyro, these assumptions lead to the performance equation:

$$I_g \frac{d^2\theta}{dt^2} + c_g \frac{d\theta}{dt} + k_g\theta = H_s\left[\omega_{IA} - \omega_{cmd} - \theta\omega_{SRA} + \frac{U\left(M_{OA}\right)}{H_s}\right] - I_g \frac{d\omega_{OA}}{dt}$$

For integrating gyros, a restraining torsional spring is eliminated, hence $k_g = 0$ and the performance equation becomes:

$$\tau_g \frac{d^2\theta}{dt^2} + \frac{d\theta}{dt} = \frac{H_s}{c_g}\left[\omega_{IA} - \omega_{cmd} - \theta\omega_{SRA} + \frac{U\left(M_{OA}\right)}{H_s}\right] - \tau_g \frac{d\omega_{OA}}{dt}$$

or $$\frac{d}{dt}\left(\tau_g \frac{d\theta}{dt} + \theta\right) = \frac{d}{dt}\left(\int \omega_{IA} - \omega_{cmd} - \theta\omega_{SRA} + \frac{U\left(M_{OA}\right)}{H_s}\, dt\right) - \tau_g\, \omega_{OA}$$

Therefore:

$$\tau_g \frac{d\theta}{dt} + \theta = \frac{H_s}{c_g}\int\left(\omega_{IA} - \omega_{cmd} - \theta\omega_{SRA} + \frac{U\left(M_{OA}\right)}{H_s}\right)dt - \tau_g\ \omega_{OA}$$

Where

$\theta \equiv$ Output axis rotation

$\omega_{IA} \equiv$ Input axis angular rate

$\omega_{cmd} \equiv$ Commanded output axis angular rate

$\omega_{SRA} \equiv$ Spin reference axis angular rate

$H_s \equiv$ Rotor angular momentum

$\tau_g \equiv I_g/c_g \equiv$ time constant

$I_g \equiv$ gyro output axis effective moment of inertia

$c_g \equiv$ float damping coefficient

$U\left(M_{OA}\right) \equiv$ Uncertain torque about output axis

Assuming θ & $\tau_g <<1$ & $\omega_{IA} = \omega_{cmd}$, the equation becomes

$$\tau_g \frac{d\theta}{dt} + \theta = \frac{1}{c_g}\int U\left(M_{OA}\right) dt$$

This equation shows that the gyro drift uncertainty is a first order response to the time integral of the uncertainty torques about the output axis.

Alternately expressing the equation in terms of drift rate:

$$\tau_g \frac{d\omega_{OA}}{dt} + \omega_{OA} = \frac{U\left(M_{OA}\right)}{c_g}$$

Hence, any source of uncertain torque of the torque summing member about the output axis is a contributor to the possible inaccuracy of the gyro element.

The most common sources of these uncertainties are gimbal friction and mass unbalance.

These are factors very sensitive to the material state and processing variables. It is for this reason that a rational method of selecting injection molding parameters is required. A brief example of this is presented.

Prior to introduction into service, the molded gyro is balanced. Remaining unbalance can be nullified by compensation in the feedback loop of the control system. However, from the drift rate equation we can see that for a step acceleration the steady state ($t \rightarrow \infty$) drift rate, due to torque uncertainties caused by variations to the balance, is:

$$\omega_{OA}\Big|_{s.s.} = \frac{\rho V e a g \tau_g}{c_g}$$

Where ρ is the mass density, V is the effective volume of unbalance, e is the amount of mass eccentricity, and a is the step acceleration in g's.

Taking typical values:

$$c_g = 20588 \ \frac{\text{dyne-sec}}{\text{cm}}$$

$$\tau_g = 0.0017 \text{ sec}$$

$$\omega_{OA}\Big|_{s.s.} = 1^{\circ}/\text{hr} = 4.8 \times 10^{-6} \text{ rad/sec}$$

$$\rho = 1.6 \text{ gm/cm}^3 \text{ (Polypheneline Sulfide)}$$

We obtain:

$$Ve = \frac{0.0363}{a} \text{ cm}^4.$$

For an acceleration of 10gs then we get:

$$Ve = 0.00363 \text{ cm}^4$$

which defines the bounds of mass unbalance which can be tolerated, for the specified performance, due to long term materials behavior (creep relaxation, non-uniform thermal strain, etc.).

BIBLIOGRAPHY

1. Zienkiewicz, O.C., _The Finite Element Method_, Chap. 22, 3rd ed., McGraw-Hill, 1977.

2. Chung, T.J., _Finite Element Analysis in Fluid Dynamics_, McGraw-Hill, 1978.

3. Bathe, K.J., _Finite Element Procedures in Engineering Analysis_, Chap. 7, Prentice Hall, 1980.

4. Oden, J.T., Zienkiewicz, O.C., Gallagher, R.H., and Taylor, C., (eds.), _Finite Elements in Fluids_, Vols. I and II, J. Wiley and Sons, 1975.

5. Taylor, C., and Hood, P., "A Numerical Solution of the Navier-Stokes Equations Using the Finite Element Technique", J. Computers and Fluids, Vol. 1, pp 73-100, Pergamon Press, 1973.

6. Finlayson, B.A., "Weighted Residual Methods and Their Relation to Finite Element Methods in Flow Problems", Chap. 1, Vol. II of Ref. 4.

7. Hughes, T.J.R., Wing, K.L., and Brooks, A., "Review of Finite Element Analysis of Incompressible Viscous Flows By the Penalty Function Formulation", J. Computational Physics, Vol. 30, No. 1, January 1979.

8. Oldroyd, J.G., "On the Formulation of Rheological Equations of State", Proc. Royal Society, A200, p. 523, 1950.

9. Roylance, D.K., "Use of 'Penalty' Finite Elements in Analysis of Polymer Melt Processing", Polymer Engineering and Science, Vol. 20, 15, 1980.

10. Bigg, D.M., "Mixing in a Single Screw Extruder", Ph.D. Thesis, University of Massachusetts, Amherst, Mass. 1973.

11. Wang, K.K. et. al., "Cornell Injection Molding Project", Proceedings of the International Conference on Polymer Processing at the Massachusetts Institute of Technology, pp. 293-314 (1977), Edited by Nam P. Suh and Nak-Ho Sung, The MIT Press, 1979.

12 Caswell, B., and Tanner, R.I., "Wirecoating Die Design Using Finite Element Methods", Proceedings of the International Conference on Polymer Processing at the Massachusetts Institute of Technology, pp. 540-556, 1977, Edited by Nam P. Suh and Nak-Ho Sung, The MIT Press, 1979.

13. Hildebrand, F.B., Methods of Applied Mathematics, 2nd ed., Prentice Hall, 1965.

14. Finlayson, B.A., "Existence of Variational Principle for the Navier-Stokes Equations", Physics of Fluids 15, pp. 963-967, 1972.

15. Chang, P.W., Patten, T.W., and Finlayson, B.A., "Collocation and Galerkin Finite Element Methods for Viscoelastic Fluid Flow. I - Description of Method and Problems with Fixed Geometry", J. Computers and Fluids, Vol. 7, pp. 267-283, Pergamon Press 1979.

16. Frecaut, G.M., "Numerical Analysis of Axissymmetric Flows", S. M. Thesis, Massachusetts Institute of Technology, June 1980.

17. Tanner, R.I., Nickell, R.E., and Bilger, R.W., "Finite Element Methods for the Solution of Some Incompressible Non-Newtonian Fluid Mechanics Problems with Free Surfaces", Computer Methods in Applied Mechanics and Engineering, 6, pp. 155-174, 1975.

18. Bercovier, M., Hasbami, Y., Gilon, Y., and Bathe, K.J., "A Finite Element Analysis Procedure for Non-Linear, Incompressible Elasticity", Annual Progress Report, Binational Science Foundation Research Contract 1933/79, Massachusetts Institute of Technology, 1980.

19. Truesdell, C., and Noll, W., "The Non-Linear Field Theories of Mechanics", Encyclopedia of Physics ed., S. Flügge, Vol. 3/3, Springer-Verlag, Berlin, 1965.

20. Truesdell, C., The Elements of Continuum Mechanics, Springer-Verlag, New York, 1965.

21. Malvern, L.E., Introduction to the Mechanics of a Continuous Medium, Prentice-Hall, 1969.

22. Flügge, W., Viscoelasticity, Springer-Verlag, New York, 1975.

23. Han, C.D., Rheology in Polymer Processing, Academic Press, 1976.

24. Bird, R.B., Hassager, O., and Abdel-Khalek, S.I., "Co-Rotational Rheological Models and the Goddard Expansion", AIChE Journal, Vol. 20, No. 6, 1974.

25. Pipkin, A.C., and Tanner, R.I., "A Survey of Theory and Experiment in Viscometric Flows of Viscoelastic Liquids", Mechanics Today, Vol. 1, 1972.

26. Middleman, S., Fundamentals of Polymer Processing, McGraw-Hill, 1977.

27. Perera, M.G.N., and Strauss, K., "Direct Numerical Solutions of the Equations for Viscoelastic Fluid Flow", Journal of Non-Newtonian Fluid Mechanics, 5, pp. 269-283, 1979.

28. Kawahara, M., and Takeuchi, N., "Mixed Finite Element Method for Analysis of Viscoelastic Fluid Flow", Computers and Fluids, Vol. 5, pp. 33-45, 1977.

29. Hildebrand, F.B., "Advanced Calculus for Applications, 2nd ed., pp. 105-107, Prentice-Hall, 1976.

30. Roache, P.J., Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, New Mexico, 1976.

31. Lax, P.D., and Richtmyer, R.D., "Survey of the Stability of Finite Difference Equations", Commentary on Pure and Applied Mathematics, 9, pp. 267-293, 1956.

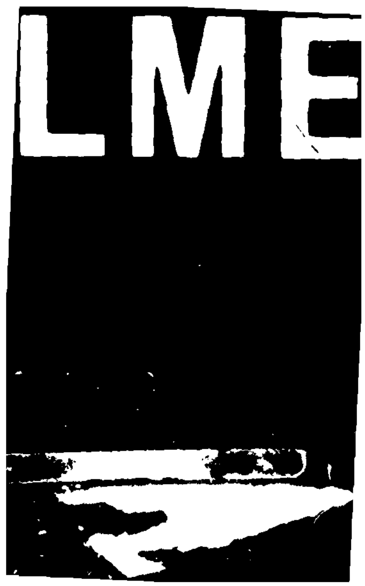
32. Engleman, M.S., Strong, G., and Bathe, K.J., "Application of Quasi-Newton Methods in Fluid Mechanics", to be published in International Journal of Numerical Methods in Engineering.

33. White, J.L., "Extrusion of Polymer Melt Systems Through Dies", Proceedings of the International Conference on Polymer Processing at the Massachusetts Institute of Technology, pp. 10-33 (1977), Edited by Nam P. Suh and Nak-Ho Sung, The MIT Press, 1979.

34. Tritton, D.J., Physical Fluid Dynamics, Van Nostrand Reinhold Co. 1977.

35. Batchelor, G.K., An Introduction to Fluid Dynamics, Cambridge University Press, Cambridge, Great Britain, 1967.

36. Wrigley, W., Hollister, W.M., and Denhard, W.G., Gyroscopic Theory, Design, and Instrumentation, The MIT Press, Cambridge, Mass. 1969.


LME